

最优化方法及其在机械行业中的应用

罗中华 著

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书共计 9 章和 1 个附录。第 1~6 章, 论述最优化理论与方法; 第 7~9 章, 阐述作者在最优化设计方面的研究实例; 最后, 附录 A 为用 C++ 语言编写的多层压配组合挤压凹模疲劳强度优化设计程序实例。

第 1 章, 阐述二次函数、梯度与 Hessian 矩阵、多元函数的 Taylor 展开、凸集与凸函数、极小点和最优解的充要条件和下降迭代法等最优化设计理论的基本概念; 第 2 章, 论述一维优化方法中的平分法、黄金分割法 (0.618 法) 和抛物线插值法; 第 3 章, 论述无约束最优化问题求解方法中的梯度法、共轭梯度法、牛顿法、变尺度法、模式搜索法、方向加速法 (Powell 法) 和单纯形法; 第 4 章, 论述约束最优化问题求解方法中的三种罚函数法、复合形法、半惩罚函数法和增广拉格朗日乘子法; 第 5 章, 讨论多目标函数的优化设计方法中的理想点法、线性加权法、乘除法和极大极小法; 第 6 章, 简介遗传算法; 第 7 章, 论述两个最优化方法在压力加工中的应用实例; 第 8 章, 论述七个最优化方法在机构设计中的应用实例; 第 9 章, 论述两个最优化方法在拟合公式中的应用实例。

本书可作为高等院校机械类本科生和研究生的教材, 对设计工程师和研究学者也颇有参考价值。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目 (CIP) 数据

最优化方法及其在机械行业中的应用 / 罗中华著. —北京: 电子工业出版社, 2008.2

ISBN 978-7-121-05337-5

I. 最… II. 罗… III. 最佳化—数学方法—应用—机械工业 IV. 0224 TH

中国版本图书馆 CIP 数据核字 (2008) 第 003908 号

策划编辑: 田领红

责任编辑: 宋兆武 陈心中

印 刷:

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 14.75 字数: 369 千字

印 次: 2008 年 2 月第 1 次印刷

印 数: 4 000 册 定价: 25.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

最优化问题可追溯到公元前 300 年左右古希腊著名的数学家欧几里得 (Euclid)，他指出在周长相同的一切矩形中，以正方形的面积为最大。17 世纪，牛顿、莱布尼兹等著名数学家创立了微积分，为最优化方法的研究提供了理论基础。

机械优化设计是计算机应用技术、最优化原理与方法和机械设计相结合的一门先进设计技术。它是最优化理论和方法为基础，以计算机为运算工具来寻求最优的机械设计参数的一种现代设计方法。

由于经济全球化，产品的竞争日趋激烈，对产品设计质量要求越来越高、设计周期要求越来越短，传统设计已经不能适应工业发展的需要。为了使机械设计师较好地掌握优化设计方法和将优化设计方法应用于机械设计中，作者在 20 多年从事最优化设计研究的基础上撰写了本书。

20 多年来，作者对最优化方法进行了不懈的研究，取得了一些研究成果。

在最优化理论和方法方面的研究成果有：①用数学中范数概念正确地证明了 n 维直交定理；②提出了一种求解最优化问题可行解的快速有效方法；③提出了半惩罚函数最优化方法。

在最优化设计方面的研究成果有：①多层压配组合挤压凹模优化设计；②冲裁零件排样优化设计；③飞剪机剪切机构的优化设计；④蟹爪式装载机扒取机构的优化设计；⑤热锯机送进机构的优化设计；⑥J88 型冷挤压机拉力肘杆机构优化设计；⑦八连杆双动拉延压力机内外滑块执行机构优化设计；⑧双动拉延压力机内滑块十杆机构的优化设计；⑨有心扇形滑移线场的近似解析式的拟合；⑩假塑性塑料熔体在鱼尾形流道中的近似流动公式的拟合等。

这些研究成果中，有些是作者首次提出来的，有些是在其他学者的研究基础上再研究的成果。在此，作者向这些研究学者表示崇高的敬意。

本书的出版得到了上海建桥学院和机电工程系的资助。在撰写过程中，实训中心的部分教师承担了部分打印工作。在此，向他们表示衷心的感谢。

由于作者水平有限，书中错误和不妥之处一定不少，恳请各位专家、学者批评指正。

罗中华
2007 年 11 月

目 录

| | |
|---------------------------------------|----|
| 第 1 章 最优化问题基础知识 | 1 |
| 1.1 最优化问题的实例 | 1 |
| 1.1.1 利润问题 | 1 |
| 1.1.2 双臂架的最优化设计 | 2 |
| 1.2 最优化问题的基本概念 | 3 |
| 1.2.1 最优化问题的一般形式 | 3 |
| 1.2.2 最优化问题的分类 | 5 |
| 1.3 有关线性代数方面的知识 | 5 |
| 1.3.1 正定矩阵 | 5 |
| 1.3.2 Euclid 空间 | 6 |
| 1.3.3 向量的范数与 Cauchy-Schwarz 不等式 | 6 |
| 1.4 二次函数 | 8 |
| 1.5 梯度与 Hessian 矩阵 | 9 |
| 1.5.1 梯度 | 9 |
| 1.5.2 Hessian 矩阵 | 11 |
| 1.6 多元函数的 Taylor 展开 | 11 |
| 1.7 凸集与凸函数 | 12 |
| 1.7.1 凸集 | 12 |
| 1.7.2 凸函数 | 13 |
| 1.8 极小点与最优解的充要条件 | 16 |
| 1.8.1 极小点的定义 | 16 |
| 1.8.2 最优解的充要条件 | 18 |
| 1.9 下降迭代法及其收敛性 | 22 |
| 1.9.1 下降迭代法 | 22 |
| 1.9.2 收敛速度与收敛准则 | 24 |
| 第 2 章 一维搜索 | 25 |
| 2.1 搜索区间的确定 | 25 |
| 2.1.1 单谷函数与搜索区间 | 25 |
| 2.1.2 确定搜索区间的进退法 | 26 |
| 2.2 平分法 | 30 |
| 2.2.1 利用函数导数的平分法 | 30 |
| 2.2.2 仅利用函数本身的平分法 | 32 |
| 2.3 黄金分割法 | 34 |
| 2.4 抛物线插值法 | 37 |
| 第 3 章 无约束最优化问题的求解方法 | 40 |
| 3.1 梯度法 | 40 |

| | | |
|-------|----------------------|-----|
| 3.2 | 共轭梯度法 | 43 |
| 3.3 | 牛顿法 | 50 |
| 3.4 | 变尺度法 | 52 |
| 3.4.1 | 变尺度法的基本思想 | 52 |
| 3.4.2 | 一族变尺度法的形成 | 54 |
| 3.4.3 | DFP 变尺度法 | 56 |
| 3.4.4 | BFGS 变尺度法 | 63 |
| 3.5 | 模式搜索法 | 63 |
| 3.6 | 方向加速法 | 66 |
| 3.7 | 单纯形法 | 78 |
| 第 4 章 | 约束最优化问题的求解方法 | 82 |
| 4.1 | 罚函数法 | 82 |
| 4.1.1 | 外点罚函数法 | 83 |
| 4.1.2 | 内点罚函数法 | 87 |
| 4.1.3 | 扩展的内点罚函数法 | 90 |
| 4.2 | 复合形法 | 91 |
| 4.3 | 半惩罚函数法 | 94 |
| 4.3.1 | 半惩罚函数法提出的原因 | 94 |
| 4.3.2 | 半惩罚函数法 | 94 |
| 4.4 | 增广拉格朗日乘子法 | 96 |
| 4.4.1 | 等式约束最优化问题 | 96 |
| 4.4.2 | 不等式约束最优化问题 | 98 |
| 4.4.3 | 一般约束最优化问题 | 99 |
| 第 5 章 | 多目标函数的优化设计方法 | 101 |
| 5.1 | 引言 | 101 |
| 5.2 | 多目标函数优化设计的求解方法 | 101 |
| 5.2.1 | 理想点法 | 101 |
| 5.2.2 | 线性加权法 | 102 |
| 5.2.3 | 乘除法 | 102 |
| 5.2.4 | 极大极小法 | 103 |
| 第 6 章 | 遗传算法简介 | 104 |
| 6.1 | 引言 | 104 |
| 6.2 | 遗传算法及其特点 | 104 |
| 6.2.1 | 遗传算法的基本思路及其特点 | 104 |
| 6.2.2 | 遗传算法的基本步骤 | 105 |
| 6.3 | 遗传算法的实现方法 | 106 |
| 6.3.1 | 编码方法 | 106 |
| 6.3.2 | 适应度函数 | 109 |
| 6.3.3 | 遗传算子 | 111 |
| 6.3.4 | 遗传算法运行参数的选择 | 115 |

| | |
|---|------------|
| 6.4 遗传算法应用实例 | 116 |
| 第7章 最优化方法在压力加工中的应用 | 119 |
| 7.1 多层压配组合挤压凹模优化设计 | 119 |
| 7.1.1 组合凹模的特点 | 119 |
| 7.1.2 整体凹模受力分析 | 120 |
| 7.1.3 组合凹模的力学分析 | 122 |
| 7.1.4 组合凹模的强度校核 | 124 |
| 7.1.5 多层压配组合挤压成形凹模的优化设计 | 127 |
| 7.2 冲裁零件排样优化设计 | 132 |
| 7.2.1 冲裁零件排样 | 132 |
| 7.2.2 冲裁零件排样优化目标函数的建立 | 134 |
| 7.2.3 条料宽度和排样单元步长的计算 | 135 |
| 7.2.4 隔离开关 GN2-10/2000 的支承条零件排样优化设计 | 138 |
| 第8章 最优化方法在机构设计中的应用 | 139 |
| 8.1 飞剪机剪切机构的优化设计 | 139 |
| 8.1.1 对飞剪机剪切机构的要求 | 140 |
| 8.1.2 飞剪机构的运动分析 | 140 |
| 8.1.3 剪切机构的优化设计数学模型的建立 | 143 |
| 8.1.4 50t 飞剪机剪切机构的优化设计 | 145 |
| 8.2 蟹爪式装载机扒取机构的优化设计 | 147 |
| 8.2.1 扒取机构 | 147 |
| 8.2.2 扒取机构的运动学分析 | 148 |
| 8.2.3 扒取机构优化设计的数学模型的建立 | 149 |
| 8.2.4 矿用蟹爪式装载机的扒取机构优化设计实例 | 153 |
| 8.3 热锯机送进机构的优化设计 | 156 |
| 8.3.1 热锯机四连杆送进机构 | 156 |
| 8.3.2 送进机构运动学分析 | 157 |
| 8.3.3 送进机构优化设计数学模型的建立 | 158 |
| 8.3.4 热锯机送进机构优化设计实例 | 160 |
| 8.4 J88 型冷挤压机拉力肘杆机构优化设计 | 161 |
| 8.4.1 冷挤压机概述 | 161 |
| 8.4.2 J88 型冷挤压机拉力肘杆机构 | 162 |
| 8.4.3 拉力肘杆机构非独立构件尺寸参数的计算 | 162 |
| 8.4.4 拉力肘杆机构的运动分析 | 164 |
| 8.4.5 拉力肘杆机构优化设计数学模型的建立 | 166 |
| 8.4.6 J88-400 型冷挤压机拉力肘杆机构优化设计 | 168 |
| 8.5 JA88 型冷挤压机压力肘杆机构优化设计 | 169 |
| 8.5.1 JA88 型冷挤压机压力肘杆机构 | 169 |
| 8.5.2 压力肘杆机构非独立构件尺寸参数的计算 | 170 |
| 8.5.3 压力肘杆机构的运动分析 | 171 |

| | | |
|-------|--|-----|
| 8.5.4 | 压力肘杆机构的受力分析与曲柄轴摩擦扭矩的计算 | 172 |
| 8.5.5 | 压力肘杆机构优化设计数学模型的建立 | 175 |
| 8.5.6 | JA88-160 型冷挤压机压力肘杆机构优化设计 | 177 |
| 8.6 | 八连杆双动拉延压力机内外滑块执行机构优化设计 | 177 |
| 8.6.1 | 板料拉延工艺 | 177 |
| 8.6.2 | 八连杆双动拉延压力机内外滑块执行机构 | 179 |
| 8.6.3 | 机构构件非独立构件尺寸参数的计算 | 179 |
| 8.6.4 | 内、外滑块位移和速度公式的推导 | 182 |
| 8.6.5 | 八连杆内、外滑块执行机构优化设计目标函数的建立 | 184 |
| 8.6.6 | 八连杆双动拉延压力机内外滑块执行机构优化设计实例 | 187 |
| 8.7 | 双动拉延压力机内滑块十杆机构的优化设计 | 189 |
| 8.7.1 | 内滑块十杆机构 | 189 |
| 8.7.2 | 非独立机构构件尺寸参数的计算 | 190 |
| 8.7.3 | 内滑块位移和速度公式的推导 | 191 |
| 8.7.4 | 内滑块十杆机构优化设计数学模型的建立 | 193 |
| 8.7.5 | 用半惩罚函数法求内滑块十杆机构优化设计的最优解 | 197 |
| 8.7.6 | 双动拉延压力机内滑块十杆机构的优化设计实例 | 197 |
| 第 9 章 | 最优化方法在拟合公式中的应用 | 199 |
| 9.1 | 有心扇形滑移线场的近似解析式的拟合 | 199 |
| 9.1.1 | 有心扇形滑移线场的近似解析式 | 199 |
| 9.1.2 | 有心扇形场的差分公式 | 202 |
| 9.1.3 | 用优化方法确定有心扇形滑移近似解析式 | 202 |
| 9.1.4 | 有心扇形滑移线场的近似解析式的应用 | 203 |
| 9.2 | 假塑性塑料熔体在鱼尾形流道中的近似流动公式的拟合 ^[53] | 204 |
| 9.2.1 | 假塑性塑料熔体在鱼尾形流道中的流动公式的推导 | 204 |
| 9.2.2 | 用优化方法确定插值公式 | 207 |
| 附录 A | 多层压配组合挤压凹模疲劳强度优化设计 C++ 程序 | 211 |
| 参考文献 | | 223 |

第 1 章 最优化问题基础知识

1.1 最优化问题的实例

1.1.1 利润问题

某煤厂生产市场上畅销的褐煤和烟煤，两种煤每生产 1 吨的利润分别为 4 元和 3 元。该煤厂的破碎、过筛和清洗分厂每天工作时间分别不超过 12 小时、10 小时和 8 小时，生产 1 吨褐煤需要 3 小时破碎、3 小时过筛和 4 小时清洗；生产 1 吨烟煤需要 4 小时破碎、3 小时过筛和 2 小时清洗。问该厂每天生产褐煤和烟煤各多少吨才能达到最大的利润。

设每天生产褐煤 x_1 吨、烟煤 x_2 吨，则每天的利润为

$$f(x_1, x_2) = 4x_1 + 3x_2$$

由于受该厂生产条件的限制，必须满足下列约束条件：

(1) 破碎工作时间不超过 12 小时，即

$$3x_1 + 4x_2 \leq 12$$

(2) 过筛工作时间不超过 10 小时，即

$$3x_1 + 3x_2 \leq 10$$

(3) 清洗工作时间不超过 8 小时，即

$$4x_1 + 2x_2 \leq 8$$

(4) x_1 和 x_2 非负，即

$$x_1, x_2 \geq 0$$

综上所述，该利润问题的最优化数学模型为

$$\max \{4x_1 + 3x_2\} \quad (1.1.1)$$

满足于

$$\begin{aligned} 3x_1 + 4x_2 &\leq 12 \\ 3x_1 + 3x_2 &\leq 10 \\ 4x_1 + 2x_2 &\leq 8 \\ x_1, x_2 &\geq 0 \end{aligned} \quad (1.1.2)$$

1.1.2 双臂架的最优化设计

考虑由空心圆管构成的对称双臂架，如图 1.1.1 所示。已知顶点承受的载荷为 $2P$ 、两支柱的跨距为 $2L$ 、圆管材料的弹性模量为 E 、材料的泊松系数为 ν 、屈服强度 σ_y 和密度为 ρ 。如何选择圆管的平均直径、壁厚和双臂架的高度，使双臂架的重量最轻（即最节省材料）。

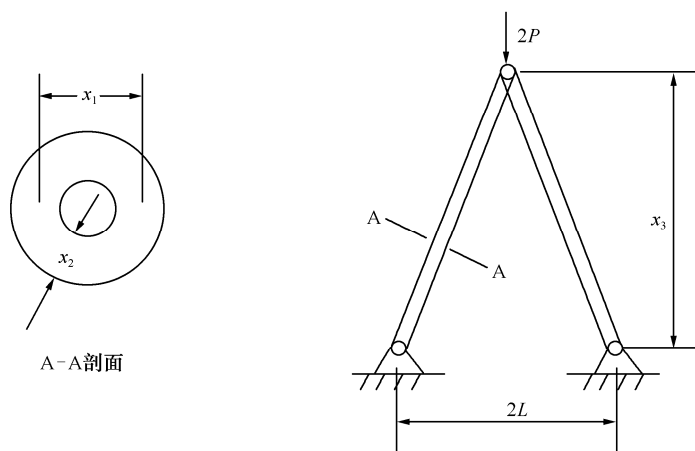


图 1.1.1 双臂架

设双臂架所使用的圆管的平均直径为 x_1 、壁厚为 x_2 和双臂架的高度 x_3 ，则双臂架的重量 W 为

$$W = 2\pi\rho x_1 x_2 \sqrt{L^2 + x_3^2}$$

必须满足以下约束条件：

(1) 圆管中的应力不超过材料的屈服强度 σ_y ，即

$$P\sqrt{L^2 + x_3^2} \leq \pi\sigma_y x_1 x_2 x_3$$

(2) 保证在载荷作用下圆管不发生失稳，由材料力学可知圆管的临界载荷为 $P_{cr} = \pi^2 EI / (L^2 + x_3^2)$ ，其中，截面惯性矩为 $I = \pi x_1 x_2 (x_1^2 + x_2^2) / 8$ 。整理后，可得

$$8P(L^2 + x_3^2)^{3/2} \leq \pi^3 E x_1 x_2 x_3 (x_1^2 + x_2^2)$$

(3) 由于空间的限制，双臂架的高度 x_3 不超过某一极限高度 h_{\max} ，即

$$x_3 \leq h_{\max}$$

(4) 圆管成立的条件为

$$x_1 - x_2 \geq d_{\min}$$

式中， d_{\min} 为制造圆管所允许的最小内径。

(5) 由于制造上的原因，圆管的平均直径与壁厚之比不得超过某一常数 c ，即

$$x_1 \leq cx_2$$

(6) 保证薄臂壳不发生局部弯曲或翘曲, 要求圆管应力小于薄臂壳不发生局部弯曲的极限应力为

$$\sigma_s = \frac{2 E x_2}{x_1 \sqrt{3(1-\nu^2)}} \quad [1]$$

则

$$P \sqrt{3(1-\nu^2)} \sqrt{L^2 + x_3^2} \leq 2\pi E x_2^2 x_3$$

(7) x_1 、 x_2 和 x_3 大于零, 即

$$x_1, x_2, x_3 > 0$$

综上所述, 双臂架的最优化设计的数学模型为

$$\min \left\{ 2\pi\rho x_1 x_2 \sqrt{L^2 + x_3^2} \right\} \quad (1.1.3)$$

满足于

$$\begin{aligned} P \sqrt{L^2 + x_3^2} &\leq \pi \sigma_y x_1 x_2 x_3 \\ 8P(L^2 + x_3^2)^{3/2} &\leq \pi^3 E x_1 x_2 x_3 (x_1^2 + x_2^2) \\ x_3 &\leq h_{\max} \\ x_1 - x_2 &\geq d_{\min} \\ x_1 &\leq c x_2 \\ P \sqrt{3(1-\nu^2)} \sqrt{L^2 + x_3^2} &\leq 2\pi E x_2^2 x_3 \\ x_1, x_2, x_3 &> 0 \end{aligned} \quad (1.1.4)$$

1.2 最优化问题的基本概念

1.2.1 最优化问题的一般形式

在研究最优化问题中, 经常将变量 x_1, x_2, \dots, x_n 视为 n 维向量空间 \mathbf{R}^n 中的一个向量 \mathbf{X} 的 n 个分量, 即

$$\mathbf{X} = \{x_1, x_2, \dots, x_n\}^T$$

最优化问题一般可表示为如下标准形式:

$$\begin{aligned} &\min f(\mathbf{X}) && (\mathbf{X} \in \mathbf{R}^n) \\ \text{满足于} &g_i(\mathbf{X}) \leq 0 && (i=1, 2, \dots, m) \\ &h_j(\mathbf{X}) = 0 && (j=1, 2, \dots, l) \end{aligned} \quad (1.2.1)$$

其他形式的最优化问题可以转化为如上最优化问题的标准形式, 例如, 求 $\max f(\mathbf{X})$ 可转化为求 $\min \{-f(\mathbf{X})\}$, 同理, $g_i(\mathbf{X}) \geq 0$ 亦可转化为标准形式 $-g_i(\mathbf{X}) \leq 0$ 。

在式 (1.2.1) 中, $f(\mathbf{X})$ 称为目标函数 (或者称为评价函数), 用它来评价一个设计方案好坏的具体的数值评判标准; $g_i(\mathbf{X}) \leq 0$ ($i=1, 2, \dots, m$) 称为不等式约束条件, m 为不等式约

束条件的个数; $h_j(\mathbf{X})=0$ ($j=1,2,\dots,l$) 称为等式约束条件, l 为等式约束条件的个数 [注: $l < n$]。不等式约束条件和等式约束条件统称为约束条件。

满足所有约束条件的向量 $\mathbf{X} \in \mathbf{R}^n$ 称为可行点 (或称为可行解), 所有可行点组成的集合称为可行解集, 可行解集用 \mathbf{D} 来表示, 则有

$$\mathbf{D} = \{ \mathbf{X}: g_i(\mathbf{X}) \leq 0 \quad (i=1,2,\dots,m); h_j(\mathbf{X})=0 \quad (j=1,2,\dots,l); \mathbf{X} \in \mathbf{R}^n \} \quad (1.2.2)$$

最优化问题的标准形式可表示为

$$\begin{aligned} \min f(\mathbf{X}) \\ \mathbf{X} \in \mathbf{D} \end{aligned} \quad (1.2.3)$$

最优化问题 (1.2.1) 的求解是指在可行解集 \mathbf{D} 中找一点 \mathbf{X}^* , 使

$$f(\mathbf{X}^*) = \min f(\mathbf{X}) \quad \text{对于所有 } \mathbf{X} \in \mathbf{D}$$

满足于

$$g_i(\mathbf{X}^*) \leq 0 \quad (i=1,2,\dots,m)$$

$$h_j(\mathbf{X}^*) = 0 \quad (j=1,2,\dots,l)$$

\mathbf{X}^* 称为最优化问题 (1.2.1) 的最优点, $f(\mathbf{X}^*)$ 称为最优值, $\{\mathbf{X}^*, f(\mathbf{X}^*)\}$ 称为最优解。

下面, 我们用利润问题来说明最优化问题的基本概念, 设计变量 $\mathbf{X} \in \mathbf{R}^2$, 则

$$\mathbf{X} = \{x_1, x_2\}^T$$

优化问题 (1.2.1) 中的目标函数

$$f(\mathbf{X}) = 4x_1 + 3x_2$$

约束条件为

$$g_1(\mathbf{X}) = 3x_1 + 4x_2 - 12 \leq 0$$

$$g_2(\mathbf{X}) = 3x_1 + 3x_2 - 10 \leq 0$$

$$g_3(\mathbf{X}) = 4x_1 + 2x_2 - 8 \leq 0$$

$$g_4(\mathbf{X}) = -x_1 \leq 0$$

$$g_5(\mathbf{X}) = -x_2 \leq 0$$

这个优化问题的可行解集为 $\mathbf{D} = \{ \{x_1, x_2\}^T : 3x_1 + 4x_2 - 12 \leq 0, \dots, -x_2 \leq 0, \mathbf{X} \in \mathbf{R}^2 \}$ 。

$\mathbf{X} = \{1, 1\}^T \in \mathbf{D}$, 就是一个可行点。利润问题的可行解集如图 1.2.1 所示。

在实际最优化问题中, 每个可行点就是一个可行的设计方案。不同的可行点对应着不同的目标函数值, 目标函数值的大小就是衡量一个设计方案的好坏的数量表征, 因此, 目标函数亦称为评价函数。

用图解法或其他方法求得上述利润问题的最优点为 $\mathbf{X}^* = \{0.8, 2.4\}^T$, 最优值为 $f(\mathbf{X}^*) = 10.4$ 。

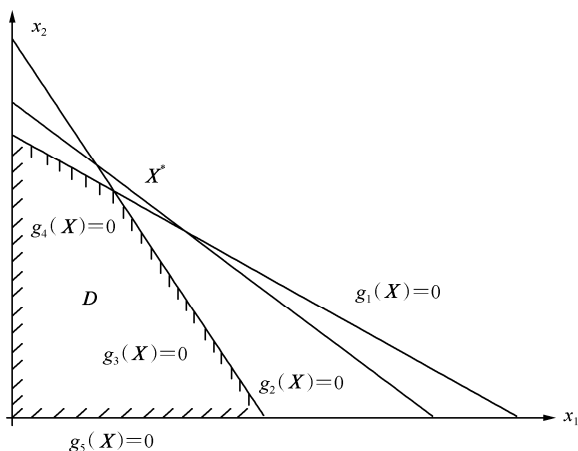


图 1.2.1 利润问题的可行解集

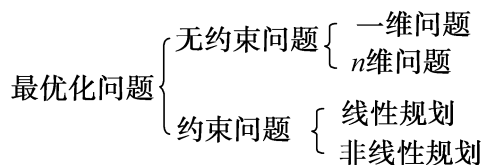
1.2.2 最优化问题的分类

在最优化问题 (1.2.1) 中, 如果 $m=l=0$, 最优化问题转化为没有约束的优化问题, 即

$$\min f(\mathbf{X}) \quad (1.2.4)$$

称为无约束问题; 当 m 和 l 不同时为零时, 称为约束问题。在约束问题中, 如果 $f(\mathbf{X})$ 、 $g_i(\mathbf{X}) \leq 0$ 和 $h_j(\mathbf{X})=0$ 都是线性函数称为线性规划, 否则称为非线性规划。上节中, 利润问题为线性规划, 双臂架最优化问题为非线性规划。在无约束问题中, 如果目标函数是一元函数称为一维问题, 如果目标函数的自变量 $\mathbf{X} \in \mathbf{R}^n$ 称为 n 维问题。

综上所述, 最优化问题的分类可归纳为



1.3 有关线性代数方面的知识

1.3.1 正定矩阵

定义 1.3.1 设 A 为 $n \times n$ 阶对称矩阵, 若对于任意向量 $\mathbf{X} \in \mathbf{R}^n$, 且 $\mathbf{X} \neq \mathbf{0}$ 向量, 都有 $\mathbf{X}^T A \mathbf{X} > 0$, 则称 A 为正定矩阵; 若对于任意向量 $\mathbf{X} \in \mathbf{R}^n$, 都有 $\mathbf{X}^T A \mathbf{X} \geq 0$, 则称 A 为半正定矩阵。若对于任意向量 $\mathbf{X} \in \mathbf{R}^n$, 且 $\mathbf{X} \neq \mathbf{0}$ 向量, 都有 $\mathbf{X}^T A \mathbf{X} < 0$, 则称 A 为负定矩阵; 若对于任意向量 $\mathbf{X} \in \mathbf{R}^n$, 都有 $\mathbf{X}^T A \mathbf{X} \leq 0$, 则称 A 为半负定矩阵。

定理 1.3.1 (Sylvester 定理) 一个 $n \times n$ 阶对称矩阵 A 为正定矩阵的充要条件是矩阵的

各阶主子式

$$\Delta_1 = a_{11}, \Delta_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, \dots, \Delta_n = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{12} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{n2} & \cdots & a_{nn} \end{vmatrix}$$

都大于零（定理的证明见文献[2]P337~P339）。

1.3.2 Euclid 空间

定义 1.3.2 设 V 是实数域 R 上的线性空间，如果在 V 上定义一个实数 (X, Y) ，它满足下列条件：

- (1) $(X, Y) = (Y, X)$
- (2) $(X + Y, Z) = (X, Z) + (Y, Z)$
- (3) $(\alpha X, Y) = \alpha(X, Y)$
- (4) $(X, X) \geq 0$ ，当且仅当 $X = 0$ 向量时， $(X, X) = 0$

这里 X, Y, Z 是 V 中的任意向量， α 为任意实数，则称这样的实数 (X, Y) 为向量 X 与 Y 的内积，而定义了内积的线性空间 V 称为 Euclid 空间。

在向量空间 R^n 中，可定义内积为

$$(X, Y) = \sum_{i=1}^n x_i y_i$$

式中 $X, Y \in R^n$ ， x_i, y_i 分别为向量 X, Y 的第 i 个元素。显然，上式满足内积定义的条件 (1)~(4)，

所以， $(X, Y) = \sum_{i=1}^n x_i y_i$ 可作为向量空间 R^n 的一种内积。

在向量空间 R^n 中，定义内积 $(X, Y) = \sum_{i=1}^n x_i y_i$ ，这时，向量空间 R^n 就成为了 Euclid 空间。

1.3.3 向量的范数与 Cauchy-Schwarz 不等式

定义 1.3.3 设 V 是数域 K 上的一个线性空间， $X \in V$ 为任意向量， X 对应一个非负的实数 $\|X\|$ ， $\|X\|$ 满足如下三个条件：

- (1) $\|X\| \geq 0$ ，当且仅当 $X = 0$ 向量时， $\|X\| = 0$
- (2) 对于任意常数 $\alpha \in K$ ， $\|\alpha X\| = |\alpha| \|X\|$
- (3) 对于任意 $X, Y \in V$ ，如下“三角不等式”成立：

$$\|X + Y\| \leq \|X\| + \|Y\| \quad (1.3.1)$$

则称实数 $\|X\|$ 为向量 X 的范数。

在向量空间 R^n 中，可定义向量 X 的范数 $\|X\|$ 为

$$\|X\| = (X, X)^{1/2} = \left(\sum_{i=1}^n x_i^2 \right)^{1/2} \quad (1.3.2)$$

在验证之前, 我们先引进 Cauchy-Schwarz 不等式。

定理 1.3.2 (Cauchy-Schwarz 不等式) 设 $X, Y \in R^n$, 则有

$$|(X, Y)| \leq \|X\| \|Y\| \quad (1.3.3)$$

证 对于任意实数 α , 恒有

$$(\alpha X + Y, \alpha X + Y) \geq 0$$

将上式展开, 得

$$\alpha^2 (X, X) + 2\alpha (X, Y) + (Y, Y) \geq 0$$

上式左边是 α 的二次三项式, 由于它的符号不变, 所以它的判别式

$$(X, Y)^2 - (X, X)(Y, Y) \leq 0$$

由上式可得

$$|(X, Y)| \leq \|X\| \|Y\|$$

证毕。

现在, 来验证由式 (1.3.2) 定义的 $\|X\| = (X, X)^{1/2} = \left(\sum_{i=1}^n x_i^2 \right)^{1/2}$ 是一种范数。显然, 它满足

向量范数的条件 (1) 和条件 (2)。下面, 证明它满足范数的“三角不等式”条件。

利用 Cauchy-Schwarz 不等式, 有

$$\begin{aligned} (X+Y, X+Y) &= (X, X) + 2(X, Y) + (Y, Y) \\ &\leq \|X\|^2 + 2\|X\|\|Y\| + \|Y\|^2 \\ &\leq (\|X\| + \|Y\|)^2 \end{aligned}$$

上式两边开平方, 得

$$\|X+Y\| \leq \|X\| + \|Y\|$$

上式说明 $\|X\| = (X, X)^{1/2} = \left(\sum_{i=1}^n x_i^2 \right)^{1/2}$ 满足范数条件 (3), 因此, 它是向量空间 R^n 中的一

种范数。

另外, 还可以通过对称正定矩阵来定义内积和范数。设 A 为 $n \times n$ 阶对称正定矩阵, 在向量空间 R^n 中, 亦可定义内积为

$$(X, Y) = X^T A Y \quad (1.3.4)$$

定义范数为

$$\|X\| = (X^T A X)^{1/2} \quad (1.3.5)$$

式中 $X, Y \in R^n$ 。

不难证明 $(X, Y) = X^T A Y$ 满足内积条件 (1) ~ (4), 因此, $(X, Y) = X^T A Y$ 可作为向量空间 R^n 中的一种内积。

显然, $\|X\| = (X^T A X)^{1/2}$ 满足范数条件 (1) 和条件 (2)。下面, 证明 $\|X\| = (X^T A X)^{1/2}$ 满

足范数的“三角不等式”条件。

由于 A 为 $n \times n$ 阶对称正定矩阵, 对于任意 $X, Y \in R^n$, α 为任意实数, 有

$$(\alpha X + Y)^T A (\alpha X + Y) \geq 0$$

由上式得

$$\alpha^2 X^T A X + 2\alpha X^T A Y + Y^T A Y \geq 0$$

上式左边是 α 的二次三项式, 由于它的符号不变, 所以, 它的判别式

$$(X^T A Y)^2 - X^T A X Y^T A Y \leq 0 \quad (1.3.6)$$

由上式得

$$|X^T A Y| \leq (X^T A X)^{1/2} (Y^T A Y)^{1/2}$$

即

$$\begin{aligned} |(X, Y)| &\leq \|X\| \|Y\| \\ \|X + Y\|^2 &= (X + Y)^T A (X + Y) \\ &= X^T A X + 2X^T A Y + Y^T A Y \\ &\leq \|X\|^2 + 2\|X\| \|Y\| + \|Y\|^2 \end{aligned} \quad (1.3.7)$$

故有

$$\|X + Y\| \leq \|X\| + \|Y\|$$

因此, $\|X\| = (X^T A X)^{1/2}$ 可作为向量空间 R^n 中的一种范数。

1.4 二次函数

在向量空间 R^n 中, 线性函数可以表示为

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^n b_i x_i + c \quad (1.4.1)$$

用向量形式表示

$$f(X) = B^T X + c \quad (1.4.2)$$

式中 $B = \{b_1, b_2, \dots, b_n\}^T$, c 为常数。

在最优化研究中, 二次函数起着重要的作用。二次函数的一般形式为

$$f(x_1, x_2, \dots, x_n) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j + \sum_{i=1}^n b_i x_i + c \quad (1.4.3)$$

式中 a_{ij} 、 b_i 和 c 为常数。用矩阵向量可表示为

$$f(X) = \frac{1}{2} X^T A X + B^T X + c \quad (1.4.4)$$

其中 $A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$, $B = \{b_1, b_2, \dots, b_n\}^T$, c 为常数。

A 为对称正定矩阵的二次函数是我们最关注的函数, 这是因为许多最优化方法都是建立在 A 为对称正定矩阵的二次函数的基础上。

例 1.4.1 把二次函数

$$f(x_1, x_2, x_3) = 3x_1^2 + x_2^2 + 2x_3^2 - 3x_1x_2 + x_1x_3 - 4x_1 + 5x_2 + 2$$

转化为矩阵向量的形式, 并验证矩阵 A 是否为正定矩阵。

解

$$f(X) = \frac{1}{2}(6x_1^2 - 3x_1x_2 + x_1x_3 - 3x_2x_1 + 2x_2^2 + 0x_2x_3 + x_3x_1 + 0x_3x_2 + 4x_3^2) - 4x_1 + 5x_2 + 2$$

$$= \frac{1}{2}(6x_1 - 3x_2 + x_3, -3x_1 + 2x_2 + 0x_3, x_1 + 0x_2 + 4x_3) \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + (-4, 5, 0) \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + 2$$

$$= \frac{1}{2}(x_1, x_2, x_3) \begin{bmatrix} 6 & -3 & 1 \\ -3 & 2 & 0 \\ 1 & 0 & 4 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + (-4, 5, 0) \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + 2$$

$$= \frac{1}{2}X^TAX + B^TX + c$$

式中 $A = \begin{bmatrix} 6 & -3 & 1 \\ -3 & 2 & 0 \\ 1 & 0 & 4 \end{bmatrix}$, $B = \begin{pmatrix} -4 \\ 5 \\ 0 \end{pmatrix}$, $X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$, $c = 2$ 。

A 的各阶主子式分别为

$$A_1 = 6 > 0, \quad A_2 = \begin{vmatrix} 6 & -3 \\ -3 & 2 \end{vmatrix} = 3 > 0, \quad A_3 = \begin{vmatrix} 6 & -3 & 1 \\ -3 & 2 & 0 \\ 1 & 0 & 4 \end{vmatrix} = 10 > 0$$

根据定理 1.3.1 知, 矩阵 A 为对称正定矩阵。

1.5 梯度与 Hessian 矩阵

1.5.1 梯度

定义 1.5.1 如果函数 $f(X)$ 的所有一阶偏导数存在, 则称

$$\nabla f(X) = \left(\frac{\partial f(X)}{\partial x_1}, \frac{\partial f(X)}{\partial x_2}, \dots, \frac{\partial f(X)}{\partial x_n} \right)^T \quad (1.5.1)$$

为函数 $f(\mathbf{X})$ 在 \mathbf{X} 处的梯度。

例 1.5.1 求线性函数 $f(\mathbf{X})=\mathbf{B}^T \mathbf{X}+c$ 的梯度。

解 把 $f(\mathbf{X})=\mathbf{B}^T \mathbf{X}+c$ 改写为 $f(\mathbf{X})=\sum_{i=1}^n b_i x_i+c$

$$\frac{\partial f(\mathbf{X})}{\partial x_j}=\frac{\partial}{\partial x_j}\left(\sum_{i=1}^n b_i x_i+c\right)=b_j \quad (j=1,2,\cdots,n)$$

所以, 线性函数的梯度为 $\nabla f(\mathbf{X})=\mathbf{B}$ 。

例 1.5.2 求二次函数 $f(\mathbf{X})=\frac{1}{2} \mathbf{X}^T \mathbf{A} \mathbf{X}+\mathbf{B}^T \mathbf{X}+c$ 的梯度, 其中 \mathbf{A} 为 $n \times n$ 阶对称矩阵。

解 把 $f(\mathbf{X})=\frac{1}{2} \mathbf{X}^T \mathbf{A} \mathbf{X}+\mathbf{B}^T \mathbf{X}+c$ 改写为 $f(\mathbf{X})=\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j+\sum_{i=1}^n b_i x_i+c$

$$\begin{aligned} \frac{\partial f(\mathbf{X})}{\partial x_k} &= \frac{\partial}{\partial x_k} \left(\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j + \sum_{i=1}^n b_i x_i + c \right) \\ &= \frac{1}{2} \sum_{i=1}^n a_{ik} x_i + \frac{1}{2} \sum_{j=1}^n a_{kj} x_j + b_k \\ &= \sum_{i=1}^n a_{ik} x_i + b_k \\ &= \mathbf{A}_k \mathbf{X} + b_k \quad (k=1,2,\cdots,n) \end{aligned}$$

式中, \mathbf{A}_k 为矩阵 \mathbf{A} 的第 k 行向量。故二次函数的梯度为

$$\nabla f(\mathbf{X})=\mathbf{A} \mathbf{X}+\mathbf{B} \quad (1.5.2)$$

为了证明梯度方向是函数具有最大变化率的方向, 首先引入方向导数的概念。

定义 1.5.2 设 \mathbf{S} 是 \mathbf{R}^n 中非空集, $f: \mathbf{S} \rightarrow \mathbf{R}^1$ 在 $\mathbf{X}_0 \in \mathbf{S}$ 处可微, \mathbf{P} 为非零向量, 且 $\|\mathbf{P}\|=1$, $\mathbf{X}_0+\lambda \mathbf{P} \in \mathbf{S}$, 对于 $\lambda>0$ 且足够小, $f(\mathbf{X})$ 在 \mathbf{X}_0 处沿 \mathbf{P} 的方向导数为

$$\frac{\partial f(\mathbf{X}_0)}{\partial \mathbf{P}}=\lim _{\lambda \rightarrow 0} \frac{f\left(\mathbf{X}_0+\lambda \mathbf{P}\right)-f\left(\mathbf{X}_0\right)}{\lambda} \quad (1.5.3)$$

下面, 我们来证明梯度方向是函数具有最大变化率的方向。令 $\varphi(\lambda)=f\left(\mathbf{X}_0+\lambda \mathbf{P}\right)$, 将 $\varphi(\lambda)$ 进行一阶 Taylor 展开, 得

$$f\left(\mathbf{X}_0+\lambda \mathbf{P}\right)=f\left(\mathbf{X}_0\right)+\lambda \nabla f\left(\mathbf{X}_0+t \lambda \mathbf{P}\right)^T \mathbf{P}$$

式中, $t \in(0,1)$ 。将上式代入式 (1.5.3) 得

$$\begin{aligned} \frac{\partial f\left(\mathbf{X}_0\right)}{\partial \mathbf{P}} &= \nabla f\left(\mathbf{X}_0\right)^T \mathbf{P}=\left\|\nabla f\left(\mathbf{X}_0\right)\right\|\left\|\mathbf{P}\right\| \cos \beta \\ &= \left\|\nabla f\left(\mathbf{X}_0\right)\right\| \cos \beta \end{aligned}$$

其中, β 是梯度 $\nabla f(\mathbf{X}_0)$ 与 \mathbf{P} 之间的夹角。

显然, 取 \mathbf{P} 为 $\nabla f(\mathbf{X}_0)$ 方向上的单位向量时, $\beta=0$, $\frac{\partial f(\mathbf{X}_0)}{\partial \mathbf{P}}$ 为最大, 且为正值。这就说明了正梯度方向是函数局部增大最快的方向。我们关心的是函数局部下降最快的方向, 负梯度方向是函数局部下降最快的方向。

1.5.2 Hessian 矩阵

定义 1.5.3 如果函数 $f: \mathbf{S} \rightarrow \mathbf{R}^1$ 有各二阶连续偏导数存在, 则称

$$\mathbf{H}(\mathbf{X}) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{X})}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{X})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{X})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(\mathbf{X})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\mathbf{X})}{\partial x_2^2} & \cdots & \frac{\partial^2 f(\mathbf{X})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{X})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\mathbf{X})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{X})}{\partial x_n^2} \end{bmatrix} \quad (1.5.4)$$

为 $f(\mathbf{X})$ 在 \mathbf{X} 处的 Hessian 矩阵。

可以证明, 若 $f(\mathbf{X})$ 的各二阶偏导数在 \mathbf{S} 上连续, 则有

$$\frac{\partial^2 f(\mathbf{X})}{\partial x_i \partial x_j} = \frac{\partial^2 f(\mathbf{X})}{\partial x_j \partial x_i} \quad (i, j=1, 2, \cdots, n)$$

因此, $f(\mathbf{X})$ 的 Hessian 矩阵 $\mathbf{H}(\mathbf{X})$ 为对称矩阵。

例 1.5.3 求二次函数 $f(\mathbf{X}) = \frac{1}{2} \mathbf{X}^T \mathbf{A} \mathbf{X} + \mathbf{B}^T \mathbf{X} + c$ 的 Hessian 矩阵。

解 由例 1.5.2, 得

$$\frac{\partial f(\mathbf{X})}{\partial x_k} = \sum_{i=1}^n a_{ki} x_i + b_k$$

$$\frac{\partial^2 f(\mathbf{X})}{\partial x_k \partial x_l} = a_{kl} \quad (k, l=1, 2, \cdots, n)$$

所以, $\mathbf{H}(\mathbf{X}) = \mathbf{A}$ 。

1.6 多元函数的 Taylor 展开

多元函数的二阶 Taylor 展开在最优化方法中应用较广, 下面给出多元函数的 Taylor 展开定理。

定理 1.6.1 (Taylor 展开定理) 设 $f: \mathbf{R}^n \rightarrow \mathbf{R}^1$ 具有二阶连续偏导数, \mathbf{P} 为 n 维向量, 则

$$f(X_0+P)=f(X_0)+\nabla f(X_0)^T P+\frac{1}{2}P^T H(X_0+\lambda P)P \quad (1.6.1)$$

其中, $\lambda \in [0,1]$ 。

证 引入辅助函数 $\varphi(t)=f(X_0+tP)$, 于是有

$$\varphi(0)=f(X_0), \quad \varphi(1)=f(X_0+P) \quad (1.6.2)$$

将 $\varphi(t)$ 在 $t=0$ 处二阶 Taylor 展开, 得

$$\varphi(t)=\varphi(0)+\varphi'(0)t+\frac{1}{2}\varphi''(\lambda t)t^2 \quad (1.6.3)$$

按复合函数求导, 不难证明

$$\varphi'(0)=\nabla f(X_0)^T P \quad (1.6.4)$$

$$\varphi''(\lambda t)=P^T H(X_0+\lambda tP)P \quad (1.6.5)$$

将上面两式代入式 (1.6.3), 并令 $t=1$, 就可以得到式 (1.6.1), 证毕。

式 (1.6.1) 的另一种表示形式为

$$f(X_0+P)=f(X_0)+\nabla f(X_0)^T P+\frac{1}{2}P^T H(X_0)P+\|P\|^3 \alpha(X_0, P) \quad (1.6.6)$$

1.7 凸集与凸函数

1.7.1 凸集

定义 1.7.1 具有某种特定性质的具体的或抽象的事物的全体称为集合。其中的成员称为这个集合的元素。

对于一个集合 S 来说, a 或者是集合 S 的一个元素, 这时我们说 a 属于 S , 记为 $a \in S$; a 或者不是集合 S 的一个元素, 我们说 a 不属于 S , 记为 $a \notin S$ 。

例如, 设 N 为全体自然数的一个集合, 则有 $2 \in N$, 而 $2.5 \notin N$ 。

又例如, 设集合 $S=\{(x,y):x>0,y>0\}$, 则有 $(2,3) \in S$, 而 $(-1,3) \notin S$ 。

定义 1.7.2 设 S 为 R^n 中一个非空的集合, 若对于 S 中任意两点的连线上所有点都属于集合 S , 则称 S 为凸集。凸集用数学形式可表示为, 对于任意 $X_1, X_2 \in S$, 若有

$$\lambda X_1 + (1-\lambda)X_2 \in S$$

其中, $\lambda \in [0,1]$, 则称 S 为凸集。若 S 不是凸集, 则称 S 为非凸集。凸集与非凸集如图 1.7.1 所示。

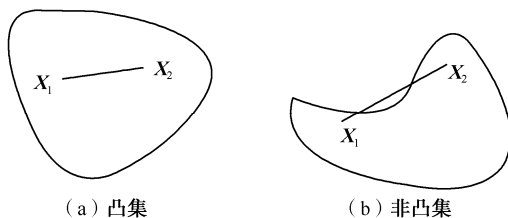


图 1.7.1 凸集与非凸集

下面是几个凸集的例子。

例 1.7.1 $S = \{(x, y) : (x-a)^2 + (y-b)^2 \leq c^2\}$

例 1.7.2 $S = \{(x_1, x_2, x_3) : x_1 + 2x_2 - x_3 = 4\}$

例 1.7.3 $S = \{X : X \text{ 是如下线性规划问题的解}\}$

$$\min C^T X$$

满足于

$$AX = B$$

$$X \geq 0$$

这里, C 是 n 维向量, B 是 m 维向量, A 是 $m \times n$ 阶矩阵, $X \in R^n$, 集合 S 是如上线性规划的最优解。

这里仅证明例 1.7.1, 其余留给读者证明。

证 设任意 $X_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$, $X_2 = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$, $X_1, X_2 \in S$, 有

$$(x_1 - a)^2 + (y_1 - b)^2 \leq c^2 \quad (1.7.1)$$

$$(x_2 - a)^2 + (y_2 - b)^2 \leq c^2 \quad (1.7.2)$$

令 $\lambda \in [0, 1]$, 则

$$X = \lambda X_1 + (1 - \lambda) X_2 = \begin{bmatrix} \lambda x_1 + (1 - \lambda) x_2 \\ \lambda y_1 + (1 - \lambda) y_2 \end{bmatrix}$$

$$\begin{aligned} & [\lambda x_1 + (1 - \lambda) x_2 - a]^2 + [\lambda y_1 + (1 - \lambda) y_2 - b]^2 \\ &= \lambda^2 [(x_1 - a)^2 + (y_1 - b)^2] + (1 - \lambda)^2 [(x_2 - a)^2 + (y_2 - b)^2] + \\ & \quad 2\lambda(1 - \lambda)[(x_1 - a)(x_2 - a) + (y_1 - b)(y_2 - b)] \end{aligned}$$

由 Cauchy-Schwarz 不等式 (1.3.3)、式 (1.7.1)、式 (1.7.2) 和上式, 得

$$[\lambda x_1 + (1 - \lambda) x_2 - a]^2 + [\lambda y_1 + (1 - \lambda) y_2 - b]^2 \leq \lambda^2 c^2 + (1 - \lambda)^2 c^2 + 2\lambda(1 - \lambda) c^2 = c^2$$

所以, $X \in S$, 根据定义 1.7.2 知, S 为凸集, 证毕。

1.7.2 凸函数

定义 1.7.3 设 S 为 R^n 中一个非空的凸集, $f: S \rightarrow R^1$ 的一个实值函数, 若对于任意 $X_1, X_2 \in S$, $\lambda \in [0, 1]$, 有

$$f[\lambda X_1 + (1 - \lambda) X_2] \leq \lambda f(X_1) + (1 - \lambda) f(X_2) \quad (1.7.3)$$

则称 f 为凸函数。若对于 S 中不同的 X_1 和 X_2 , $\lambda \in [0, 1]$, 若式 (1.7.3) 为严格的不等式, 即

$$f[\lambda X_1 + (1 - \lambda) X_2] < \lambda f(X_1) + (1 - \lambda) f(X_2) \quad (1.7.4)$$

成立, 则称 f 为严格的凸函数。如果 $-f$ 为凸函数 (或严格的凸函数), 则 f 称为凹函数 (或严格的凹函数)。

凸函数的几何解释是在凸集 S 中任意两点之间的函数值在连接该两点函数值的直线的下方。一维凸函数和凹函数如图 1.7.2 所示。

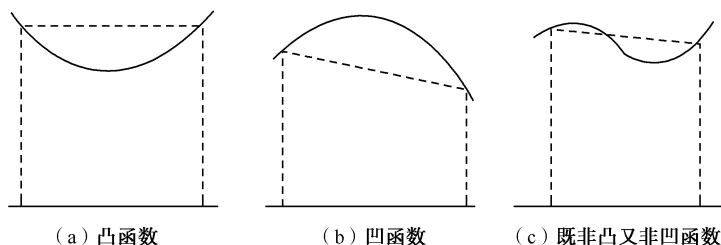


图 1.7.2 凸函数和凹函数

凸函数具有如下基本性质:

性质 1 设 f_1 和 f_2 为定义在凸集 S 上的凸函数, 则函数 f_1+f_2 在 S 上亦为凸函数。

性质 2 设 f 为定义在凸集 S 上的凸函数, 则对于任意实数 $\alpha \geq 0$, 则函数 αf 在 S 上亦为凸函数。

由以上两条性质容易推出有限个凸函数的线性组合

$$\begin{aligned} & \alpha_1 f_1 + \alpha_2 f_2 + \cdots + \alpha_m f_m \\ & \alpha_i \geq 0 \quad (i=1, 2, \dots, m) \end{aligned}$$

在 S 上亦为凸函数。

性质 3 若 X_1 和 X_2 为凸函数 $f(X)$ 中的两个最小点, 则其连线上所有点都是 $f(X)$ 的最小点。

证 由性质 3 的条件可知, $f(X_1)=f(X_2)$ 和 $f(X) \geq f(X_1)$, 设 $X = \lambda X_1 + (1-\lambda)X_2$, $\lambda \in [0, 1]$ 。因为 $f(X)$ 为凸函数, 有

$$\begin{aligned} f(X) &= f[\lambda X_1 + (1-\lambda)X_2] \\ &\leq \lambda f(X_1) + (1-\lambda)f(X_2) = f(X_1) \end{aligned}$$

由上式和 $f(X) \geq f(X_1)$ 可得 $f(X) = f(X_1)$, 故结论正确, 证毕。

在实际应用中, 如何判断一个函数是否为凸函数, 一般来说, 不是那么容易的事。在判断时, 我们可以直接利用凸函数的定义 1.7.3 去判断, 亦可以利用凸函数的性质和如下两条定理去判断。

定理 1.7.1 设 S 为凸集, $f: S \rightarrow R^1$ 具有一阶连续可微, 则 $f(X)$ 在 S 上为凸函数的充要条件是对于所有 $X, Y \in S$, 有

$$f(Y) \geq f(X) + \nabla f(X)^T (Y - X) \quad (1.7.5)$$

证 必要性 (由条件 \rightarrow 结论)

由于 $f(X)$ 为凸函数, 设任意 $X, Y \in S$ 和 $\lambda \in [0, 1]$, 有

$$f[\lambda Y + (1-\lambda)X] \leq \lambda f(Y) + (1-\lambda)f(X)$$

于是得

$$\frac{f[X + \lambda(Y - X)] - f(X)}{\lambda} \leq f(Y) - f(X)$$

将 $f[X + \lambda(Y - X)]$ 进行一阶 Taylor 展开, 并令 $\lambda \rightarrow 0$, 得

$$f(Y) \geq f(X) + \nabla f(X)^T(Y - X)$$

上式说明必要性成立。

充分性 (由结论→条件)

设任意 $X, Y \in S$, 令 $Z = \lambda X + (1 - \lambda)Y$, $\lambda \in [0, 1]$, 由于 S 为凸集, 有 $Z \in S$ 。

由式 (1.7.5), 有

$$f(X) \geq f(Z) + \nabla f(Z)^T(X - Z)$$

$$f(Y) \geq f(Z) + \nabla f(Z)^T(Y - Z)$$

由上面两式和 $Z = \lambda X + (1 - \lambda)Y$, 可得

$$\lambda f(X) + (1 - \lambda)f(Y) \geq f[\lambda X + (1 - \lambda)Y]$$

由上式和凸函数的定义 1.7.3 得知, $f(X)$ 为凸函数, 证毕。

图 1.7.3 为凸函数的定义与定理 1.7.1 不同的几何特征。按凸函数的定义是凸函数任意两点之间的函数值在连接该两点函数值的直线的下方; 而按凸函数定理 1.7.1 是凸函数的函数值在任意一点 $X \in S$ 处的切平面的上方。

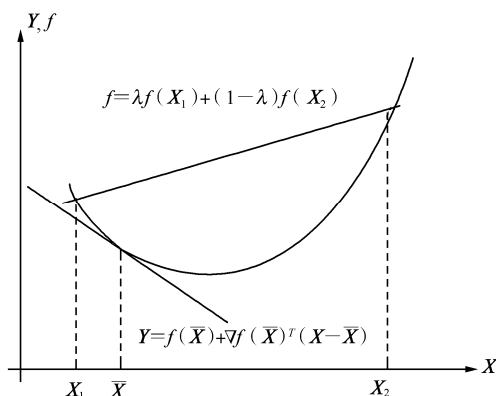


图 1.7.3 凸函数的几何特征

定理 1.7.2 设 S 为凸集, $f: S \rightarrow \mathbf{R}^1$ 具有二阶连续可微, 则 $f(X)$ 在 S 上为凸函数的充要条件是, 对于所有 $X \in S$, $f(X)$ 的 Hessian 矩阵 $H(X)$ 是半正定矩阵。

证 充要性 (由结论→条件)

设任意 $X, Y \in S$, 由于 S 为凸集, 对于 $\lambda \in [0, 1]$, 有 $X + \lambda(Y - X) \in S$, 将 $f(Y) = f[X + (Y - X)]$ 在 X 处 Taylor 展开, 得

$$f(Y) = f(X) + \nabla f(X)^T(Y - X) + \frac{1}{2}(Y - X)^T H[X + \lambda(Y - X)](Y - X) \quad (1.7.6)$$

由于 $H[X + \lambda(Y - X)]$ 为正半定矩阵, 上式左边第三项大于或等于零, 故有

$$f(Y) \geq f(X) + \nabla f(X)^T(Y - X)$$

由定理 1.7.1 知, $f(X)$ 在凸集 S 上为凸函数。

必要性 (由条件→结论)

用反证法来证明, 设有 $X \in S$ 的一个内点, 假设 $H(X)$ 为负定矩阵或不定矩阵。一定有

一点 $Y \in S$, 且 $Y \neq X$, 有

$$(Y - X)^T H(X)(Y - X) < 0$$

由于二阶偏导数的连续性, 总可取到这样的 $Y \in S$, $Y \neq X$, 对于任意 $\lambda \in [0, 1]$, 有

$$(Y - X)^T H[X + \lambda(Y - X)](Y - X) < 0$$

由上式和式 (1.7.6) 得

$$f(Y) < f(X) + \nabla f(X)^T (Y - X)$$

上式与 $f(X)$ 为凸函数相矛盾, 故必要性成立, 证毕。

例 1.7.4 证明 $f(x) = x^2 - 2x$ 在 R^1 上为凸函数。

证 按凸函数的定义 1.7.3 证明。设任意 $x_1, x_2 \in R^1$, 对于 $\lambda \in [0, 1]$, 有 $\lambda x_1 + (1 - \lambda)x_2 \in R^1$,

有

$$f(x_1) = x_1^2 - 2x_1$$

和

$$f(x_2) = x_2^2 - 2x_2$$

$$\begin{aligned} f[\lambda x_1 + (1 - \lambda)x_2] &= [\lambda x_1 + (1 - \lambda)x_2]^2 - 2[\lambda x_1 + (1 - \lambda)x_2] \\ &= \lambda^2 x_1^2 + 2\lambda(1 - \lambda)x_1 x_2 + (1 - \lambda)^2 x_2^2 - 2\lambda x_1 - 2(1 - \lambda)x_2 \\ &= \lambda(x_1^2 - 2x_1) + (1 - \lambda)(x_2^2 - 2x_2) - \lambda(1 - \lambda)x_1^2 + 2\lambda(1 - \lambda)x_1 x_2 - \lambda(1 - \lambda)x_2^2 \\ &= \lambda f(x_1) + (1 - \lambda)f(x_2) - \lambda(1 - \lambda)(x_1 - x_2)^2 \end{aligned}$$

对于不同的 x_1, x_2 和 $\lambda \in (0, 1)$, 有 $\lambda(1 - \lambda)(x_1 - x_2)^2 > 0$, 由上式可得

$$f[\lambda x_1 + (1 - \lambda)x_2] < \lambda f(x_1) + (1 - \lambda)f(x_2)$$

故 $f(x) = x^2 - 2x$ 为严格的凸函数, 证毕。

例 1.7.5 线性函数 $f(X) = B^T X + c$ 既是凸函数又是凹函数。

证 首先证明线性函数为凸函数, 设 $X_1, X_2 \in S$, 令 $X = \lambda X_1 + (1 - \lambda)X_2$, $\lambda \in [0, 1]$, 有

$$\begin{aligned} f(X) &= f[\lambda X_1 + (1 - \lambda)X_2] \\ &= \lambda B^T X_1 + (1 - \lambda)B^T X_2 + c \\ &= \lambda f(X_1) + (1 - \lambda)f(X_2) \end{aligned}$$

故 $f(X) = B^T X + c$ 为凸函数。同理可证 $-f(X)$ 为凸函数, 故 $f(X) = B^T X + c$ 为凹函数, 证毕。

1.8 极小点与最优解的充要条件

1.8.1 极小点的定义

定义 1.8.1 对于任意给定的实数 $\delta > 0$, X_0 和 $X \in R^n$, 我们称

$$N(X_0, \delta) = \{X: \|X - X_0\| < \delta, \delta > 0\}$$

为 X_0 处的一个 δ 邻域。

定义 1.8.2 设 S 为 R^n 中一个非空的凸集, $f: S \rightarrow R^1$ 上的一个实值函数, $X^* \in S$, 如果存在一个 $\delta > 0$, 使对于所有的 $X \in N(X^*, \delta) \cap S$, 都有

$$f(X^*) \leq f(X) \quad (1.8.1)$$

成立, 则称 X^* 为函数 $f(X)$ 的局部极小点。

当式 (1.8.1) 为严格的不等式时, 则称 \mathbf{X}^* 为函数 $f(\mathbf{X})$ 的严格的局部极小点。

定义 1.8.3 设 $\mathbf{X}^* \in \mathbf{S}$, 若对于一切 $\mathbf{X} \in \mathbf{S}$, 均有式 (1.8.1) 成立, 则称 \mathbf{X}^* 为函数 $f(\mathbf{X})$ 全局极小点。当对于一切 $\mathbf{X} \in \mathbf{S}$, 且 $\mathbf{X} \neq \mathbf{X}^*$, 式 (1.8.1) 为严格的不等式时, 则称 \mathbf{X}^* 为函数 $f(\mathbf{X})$ 严格的全局极小点。

现在, 叙述目标函数的等值面(线)的概念。对于给定函数 $f(\mathbf{X})$ 的值为常数 C , 设空间 \mathbf{R}^n 中有无限多点 \mathbf{X} 的函数值与之相对应。一般情况下, 函数值 $f(\mathbf{X})=C$ 与这些对应的 \mathbf{X} 在 $n+1$ 维空间 $[\mathbf{X}, f(\mathbf{X})]$ 中组成一个曲面或超曲面, 这个曲面或超曲面称为目标函数的等值面。如果 $\mathbf{X} \in \mathbf{R}^2$, 函数值 $f(\mathbf{X})=C$ 与这些对应的 \mathbf{X} 在三维空间 $[\mathbf{X}, f(\mathbf{X})]$ 中组成一条曲线, 这条曲线称为目标函数的等值线。例如, 二维函数 $z(x, y)=xe^{-x^2-y^2}$ 的曲面和等值线如图 1.8.1 所示, 在函数等值线图中, 标注为 0.2 的曲线为函数 $z(x, y)=0.2$ 的等值线, 也就是 $z=0.2$ 的平面与函数曲面的交线。

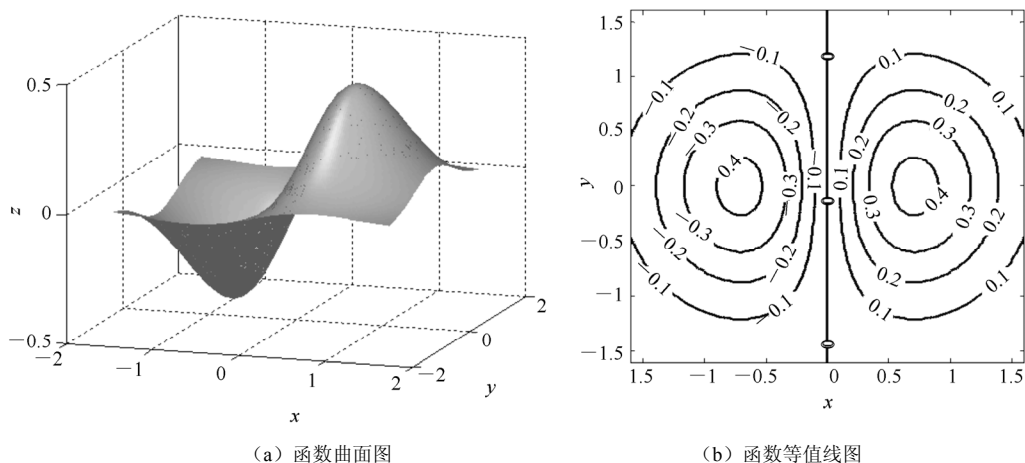


图 1.8.1 函数曲面和函数等值线

下面就一维和二维问题的极小点的定义进行几何解释。

函数 $f(x)=x \sin x$ 在区间 $S=[0, 4\pi]$ 上的图形如图 1.8.2 所示。 $x_1^*=1.5639183\pi$ 是 $f(\mathbf{X})$ 的局部极小点, $x_2^*=3.528636\pi$ 是 $f(\mathbf{X})$ 的全局极小点。

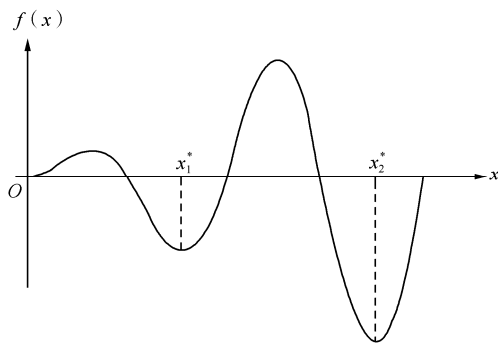


图 1.8.2 一维问题极小点的几何解释

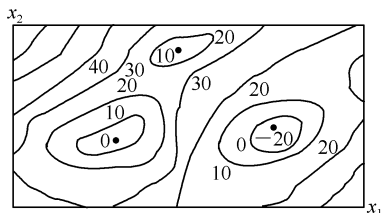


图 1.8.3 二维问题极小点的几何解释

二维问题如图 1.8.3 所示, 图中所画的曲线为曲面 $y=f(x_1, x_2)$ 的等值线, 标以 10 的曲线表示平面 $y=10$ 与曲面 $y=f(x_1, x_2)$ 的截口。显然, 图中 -20 所对应的点为 $f(x_1, x_2)$ 的全局极小点, 而 0 和 10 所对应的点为 $f(x_1, x_2)$ 的局部极小点。

1.8.2 最优解的充要条件

函数 $f(\mathbf{X})$ 的局部极小点的必要与充分条件, 以及在什么条件下, 局部极小点是全局极小点, 这是我们最关心的问题。下面, 我们分别叙述无约束问题和约束问题最优解的充要条件。

1. 无约束问题最优解的充要条件

定理 1.8.1 (必要条件) 设 $f: \mathbf{R}^n \rightarrow \mathbf{R}^1$ 在 \mathbf{X}^* 处二次可微, 如果 \mathbf{X}^* 是局部极小点, 则 $\nabla f(\mathbf{X}^*)=0$ 和 $\mathbf{H}(\mathbf{X}^*)$ 为正半定矩阵。

证 设 \mathbf{P} 是任意单位向量, 即 $\|\mathbf{P}\|=1$ 。因为 \mathbf{X}^* 是局部极小点, 所以存在一个 $\delta > 0$, 使 $\mathbf{X}^* + \lambda \mathbf{P} \in N(\mathbf{X}^*, \delta)$ 时, 总有

$$f(\mathbf{X}^* + \lambda \mathbf{P}) \geq f(\mathbf{X}^*) \quad (1.8.2)$$

引入辅助函数

$$\varphi(\lambda) = f(\mathbf{X}^* + \lambda \mathbf{P})$$

由上式和式 (1.8.2) 得

$$\varphi(\lambda) \geq \varphi(0)$$

所以, $\lambda=0$ 是 $\varphi(\lambda)$ 的局部极小点, 有

$$\varphi'(0) = 0$$

由复合函数求导, 得

$$\varphi'(0) = \nabla f(\mathbf{X}^*)^T \mathbf{P} = 0$$

由于 \mathbf{P} 的任意性, 有

$$\nabla f(\mathbf{X}^*) = 0$$

把 $f(\mathbf{X}^* + \lambda \mathbf{P})$ 在 \mathbf{X}^* 处进行 Taylor 展开

$$f(\mathbf{X}^* + \lambda \mathbf{P}) = f(\mathbf{X}^*) + \frac{1}{2} \lambda^2 \mathbf{P}^T \mathbf{H}(\mathbf{X}^*) \mathbf{P} + \lambda^3 \|\mathbf{P}\|^3 \alpha(\mathbf{X}^*, \lambda)$$

由上式得

$$\frac{f(\mathbf{X}^* + \lambda \mathbf{P}) - f(\mathbf{X}^*)}{\lambda^2} = \frac{1}{2} \mathbf{P}^T \mathbf{H}(\mathbf{X}^*) \mathbf{P} + \lambda \|\mathbf{P}\|^3 \alpha(\mathbf{X}^*, \lambda) \geq 0$$

令 $\lambda \rightarrow 0$, 得

$$\mathbf{P}^T \mathbf{H}(\mathbf{X}^*) \mathbf{P} \geq 0$$

由上式知, $\mathbf{H}(\mathbf{X}^*)$ 为正半定矩阵, 证毕。

定理 1.8.2 (充分条件) 设 $f: \mathbf{R}^n \rightarrow \mathbf{R}^1$ 具有连续二次可微, 若 $\nabla f(\mathbf{X}^*)=0$ 和 $H(\mathbf{X}^*)$ 为正定矩阵, 则 \mathbf{X}^* 为 $f(\mathbf{X})$ 的严格的局部极小点。

证 设 $N(\mathbf{X}^*, \delta)$ 为 \mathbf{X}^* 处的一个 $\delta > 0$ 邻域, 对于任意 $\mathbf{X} \in N(\mathbf{X}^*, \delta)$, 总可表示为

$$\mathbf{X} = \mathbf{X}^* + \lambda \mathbf{Y}$$

其中, \mathbf{Y} 为任意的单位向量, 即 $\|\mathbf{Y}\|=1$; λ 为实数, 且 $|\lambda| < \delta$ 。

对这样取定的 \mathbf{Y} , 定义如下函数:

$$\varphi(\lambda) = f(\mathbf{X}^* + \lambda \mathbf{Y})$$

由定理的条件可知 $\varphi(\lambda)$ 为二次连续可微函数, 且满足 $\varphi'(0)=0$ 和 $\varphi''(0)=\mathbf{Y}^T H(\mathbf{X}^*) \mathbf{Y} > 0$, 于是由 Taylor 展开可得

$$\begin{aligned} \varphi(\lambda) &= \varphi(0) + \varphi'(0)\lambda + \frac{1}{2}\varphi''(\theta\lambda)\lambda^2 \\ &= \varphi(0) + \frac{1}{2}\varphi''(\theta\lambda)\lambda^2 \end{aligned} \quad (1.8.3)$$

其中, $\theta \in (0, 1)$ 。

由连续性和 $\varphi''(0) > 0$ 可知, 存在 $\varepsilon > 0$, 对于一切 $\theta \in (0, 1)$ 和满足 $|\lambda| < \varepsilon$ 的一切 λ , 有

$$\varphi''(\theta\lambda) > 0 \quad (1.8.4)$$

由式 (1.8.3) 和式 (1.8.4) 可知, $\lambda=0$ 是 $\varphi(\lambda)$ 的严格的局部极小点。由于 \mathbf{Y} 的任意性, 由此得到 \mathbf{X}^* 为 $f(\mathbf{X})$ 的严格的局部极小点。故结论正确, 证毕。

定理 1.8.3 设 $f: \mathbf{S} \rightarrow \mathbf{R}^1$ 上连续可微的凸函数, \mathbf{S} 是 \mathbf{R}^n 中非空的凸集, \mathbf{X}^* 为 \mathbf{S} 的一个内点, 且满足 $\nabla f(\mathbf{X}^*)=0$, 则 \mathbf{X}^* 为 $f(\mathbf{X})$ 在 \mathbf{S} 上的全局极小点。

证 由于 $f(\mathbf{X})$ 为凸函数, 由定理 1.7.1 可知, 对于任意 $\mathbf{Y} \in \mathbf{S}$, $\mathbf{Y} \neq \mathbf{X}^*$, 有

$$f(\mathbf{Y}) \geq f(\mathbf{X}^*) + \nabla f(\mathbf{X}^*)^T (\mathbf{Y} - \mathbf{X}^*) = f(\mathbf{X}^*)$$

由上式知, \mathbf{X}^* 为 $f(\mathbf{X})$ 在 \mathbf{S} 上的全局极小点。

定理 1.8.4 设 \mathbf{S} 为凸集, $f: \mathbf{S} \rightarrow \mathbf{R}^1$ 上的凸函数, 则函数 $f(\mathbf{X})$ 的任一局部极小点必为全局极小点。

证 用反证法来证明, 设 $\mathbf{X}^* \in \mathbf{S}$ 为 $f(\mathbf{X})$ 的局部极小点, 若它不是全局极小点, 则必存在 $\mathbf{X}^{**} \in \mathbf{S}$, 使 $f(\mathbf{X}^{**}) < f(\mathbf{X}^*)$ 。由于 \mathbf{S} 为凸集, 有 $\mathbf{X} = \lambda \mathbf{X}^* + (1-\lambda)\mathbf{X}^{**} \in \mathbf{S}$, $\lambda \in [0, 1]$, 由于 $f(\mathbf{X})$ 为凸函数, 故有

$$\begin{aligned} f(\mathbf{X}) &= f[\lambda \mathbf{X}^* + (1-\lambda)\mathbf{X}^{**}] \\ &\leq \lambda f(\mathbf{X}^*) + (1-\lambda)f(\mathbf{X}^{**}) < f(\mathbf{X}^*) \end{aligned}$$

当 $\lambda \rightarrow 1$ 时, 上式与 \mathbf{X}^* 是 $f(\mathbf{X})$ 的局部极小点相矛盾, 故 \mathbf{X}^* 是 $f(\mathbf{X})$ 的全局极小点, 证毕。

2. 约束问题局部极小点的充要条件

定理 1.8.5 (Kuhn-Tucker) 必要条件 考虑如下最优化问题:

$$\begin{aligned} \min & f(\mathbf{X}) \\ \text{满足于} & \quad g_i(\mathbf{X}) \leq 0 \quad (i=1, 2, \dots, m) \\ & \quad h_j(\mathbf{X}) = 0 \quad (j=1, 2, \dots, l) \end{aligned}$$

设 \mathbf{X}^* 是一个可行点, $I = \{i: g_i(\mathbf{X}^*) = 0\}$, 假设 f 和 $g_i (i \in I)$ 在 \mathbf{X}^* 处可微, $g_i (i \notin I)$ 在 \mathbf{X}^* 处连续, $h_j (j=1, 2, \dots, l)$ 在 \mathbf{X}^* 处连续可微, 进一步假设 $\nabla g_i (i \in I)$ 和 $\nabla h_j (j=1, 2, \dots, l)$,

线性无关。如果 \mathbf{X}^* 是问题的局部最优解, 则存在一组 $\lambda_i, i \in I$ 和 $\lambda_{m+j} (j=1, 2, \dots, l)$, 使

$$\nabla f(\mathbf{X}^*) + \sum_{i \in I} \lambda_i \nabla g_i(\mathbf{X}^*) + \sum_{j=1}^l \lambda_{m+j} \nabla h_j(\mathbf{X}^*) = 0 \quad (1.8.5)$$

$$\lambda_i \geq 0, \quad i \in I$$

成立。如果 $g_i, i \notin I$ 亦在 \mathbf{X}^* 处可微, 上式的等价形式为

$$\nabla f(\mathbf{X}^*) + \sum_{i=1}^m \lambda_i \nabla g_i(\mathbf{X}^*) + \sum_{j=1}^l \lambda_{m+j} \nabla h_j(\mathbf{X}^*) = 0 \quad (1.8.6)$$

$$\lambda_i g_i(\mathbf{X}^*) = 0$$

$$\lambda_i \geq 0 \quad (i \in I)$$

定理的证明参看文献[3]P146, 下面, 我们仅对定理作一些几何解释。

如果 \mathbf{P} 为函数 $f(\mathbf{X})$ 在 \mathbf{X} 处函数局部下降的方向, 则有

$$\nabla f(\mathbf{X})^T \mathbf{P} < 0 \quad (1.8.7)$$

在满足 Kuhn-Tucker 条件的点 \mathbf{X}^* 处, 没有一个方向 \mathbf{P} 既使函数值局部有所下降, 又不破坏约束条件。

由式 (1.8.5) 得

$$\nabla f(\mathbf{X}^*)^T \mathbf{P} = - \sum_{i \in I} \lambda_i \nabla g_i(\mathbf{X}^*)^T \mathbf{P} - \sum_{j=1}^l \lambda_{m+j} \nabla h_j(\mathbf{X}^*)^T \mathbf{P} \quad (1.8.8)$$

(1) 选择 \mathbf{P} 使函数值局部下降

由式 (1.8.7) 和式 (1.8.8) 得

$$\sum_{i \in I} \lambda_i \nabla g_i(\mathbf{X}^*)^T \mathbf{P} + \sum_{j=1}^l \lambda_{m+j} \nabla h_j(\mathbf{X}^*)^T \mathbf{P} > 0 \quad (1.8.9)$$

① \mathbf{P} 保证等式约束不破坏, 要求:

$$\nabla h_j(\mathbf{X}^*)^T \mathbf{P} = 0 \quad (j=1, 2, \dots, l)$$

于是, 由式 (1.8.9) 有

$$\sum_{i \in I} \lambda_i \nabla g_i(\mathbf{X}^*)^T \mathbf{P} > 0$$

上式说明了至少有一个 k , 使得

$$\nabla g_k(\mathbf{X}^*)^T \mathbf{P} > 0$$

上式说明 $g_k(\mathbf{X}^*) \leq 0$ 的约束条件被破坏。

② 同理可得, 保证不等式约束不受破坏, 等式约束会破坏。

(2) 选择 \mathbf{P} , 保证不破坏约束条件

若 \mathbf{P} 不破坏约束条件, 要求:

$$\nabla h_j(\mathbf{X}^*)^T \mathbf{P} = 0 \quad (j=1, 2, \dots, l)$$

$$\nabla g_i(\mathbf{X}^*)^T \mathbf{P} \leq 0 \quad (i \in I)$$

由上面两式和式 (1.8.8) 得

$$\nabla f(X^*)^T P \geq 0$$

上式说明 P 方向不会使函数局部下降。

定理 1.8.5 的几何解释是函数的负梯度方向落在起作用的约束的梯度所组成的锥中, 二维的情况如图 1.8.4 所示。

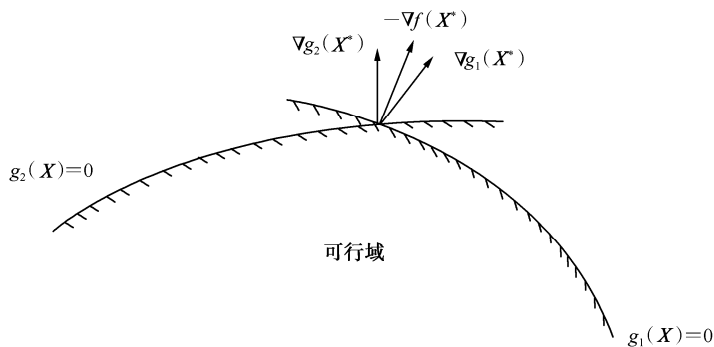


图 1.8.4 Kuhn-Tucker 条件几何解释

满足 Kuhn-Tucker 必要条件的点称为 Kuhn-Tucker 点。

例 1.8.1 考虑如下问题:

满足于

$$\begin{aligned} \min & (x_1 - 3)^2 + (x_2 - 2)^2 \\ & x_1^2 + x_2^2 - 5 \leq 0 \\ & -x_1 \leq 0 \\ & -x_2 \leq 0 \\ & x_1 + 2x_2 - 4 = 0 \end{aligned}$$

试验证 $X^* = (2, 1)^T$ 是 Kuhn-Tucker 点。

解

$$I = \{i: g_i(X^*) = 0\} = \{1\}$$

$$\nabla f(X^*) = \begin{pmatrix} 2(x_1 - 3) \\ 2(x_2 - 2) \end{pmatrix} = \begin{pmatrix} -2 \\ -2 \end{pmatrix}$$

$$\nabla g_1(X^*) = (4, 2)^T$$

$$\nabla h_1(X^*) = (1, 2)^T$$

显然, $\nabla g_1(X^*)$ 和 $\nabla h_1(X^*)$ 是线性无关的。

$$\nabla f(X^*) + \lambda_1 \nabla g_1(X^*) + \lambda_4 \nabla h_1(X^*) = 0$$

$$\begin{pmatrix} -2 \\ -2 \end{pmatrix} + \lambda_1 \begin{pmatrix} 4 \\ 2 \end{pmatrix} + \lambda_4 \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

解上式可得 $\lambda_1 = \frac{1}{3} > 0$, $\lambda_4 = \frac{2}{3}$ 。

所以, 当 $\lambda_1 = \frac{1}{3}, \lambda_4 = \frac{2}{3}$ 时, X^* 满足 Kuhn-Tucker 必要条件式 (1.8.5), 故 $X^* = (2, 1)^T$ 是

Kuhn-Tucker 点。

定理 1.8.6 (Kuhn-Tucker 充要条件)

设 $f: R^n \rightarrow R^1$; $g_i: R^n \rightarrow R^1$ ($i=1, 2, \dots, m$); $h_j: R^n \rightarrow R^1$ ($j=1, 2, \dots, l$) 的可微函数, 考虑如下最优化问题:

$$\min f(X)$$

满足于

$$g_i(X) \leq 0 \quad (i=1, 2, \dots, m)$$

$$h_j(X) = 0 \quad (j=1, 2, \dots, l)$$

设 X^* 是一个可行点, $I = \{i: g_i(X^*) = 0\}$, 进一步假设 $f, g_i(X)$ 为凸函数, $h_j(X)$ 为凹函数, 如果 X^* 是 Kuhn-Tucker 点, 即存在一组 $\lambda_i \geq 0, i \in I$ 和 λ_{m+j} ($j=1, 2, \dots, l$), 使

$$\nabla f(X^*) + \sum_{i \in I} \lambda_i \nabla g_i(X^*) + \sum_{j=1}^l \lambda_{m+j} \nabla h_j(X^*) = 0$$

成立, 则 X^* 是最优化问题的全局极小点。

定理的证明请参看文献[3]P147。

1.9 下降迭代法及其收敛性

求解无约束问题

$$\min f(X) \quad (X \in R^n) \quad (1.9.1)$$

根据定理 1.8.1, 可先求方程组

$$\nabla f(X) = 0 \quad (1.9.2)$$

的解, 即先求函数 $f(X)$ 的驻点, 然后, 再利用最优解的充分条件进行判断。对一般函数 $f(X)$ 来说, 求解式 (1.9.2) 和判断驻点是否为函数的最优点往往非常困难, 有时甚至无法实现。此外, 很多实际问题求不出或难以求得目标函数的梯度。为了避免直接求解的困难, 我们采用下降迭代法求解问题 (1.9.1)。

1.9.1 下降迭代法

首先, 给出目标函数 $f(X)$ 极小点 X^* 的一个初始估计点 X_0 , 然后, 通过某一种算法产生一个序列 $\{X_k\}$, 期望序列 $\{X_k\}$ 的极限收敛于目标函数 $f(X)$ 极小点 X^* , 即

$$\lim_{k \rightarrow \infty} X_k = X^* \quad (1.9.3)$$

或其等价形式

$$\lim_{k \rightarrow \infty} (X_k - X^*) = 0 \quad (1.9.4)$$

如果某种算法产生的序列 $\{X_k\}$, 总有 $f(X_{k+1}) < f(X_k)$, 我们称此算法为下降迭代法。

假定我们已经求得第 k 次迭代点 \mathbf{X}_k ，下一次迭代将是如下两种情况之一：

(1) 从 \mathbf{X}_k 出发沿任意方向移动，目标函数不再下降。根据定义， \mathbf{X}_k 是局部极小点，迭代终止。例如，用共轭梯度法求解二次函数，经过 n 次一维搜索能求得最优解 \mathbf{X}^* 就属于这种情况。当然，对于一般函数这种情况很少发生。

(2) 从 \mathbf{P}_k 出发至少存在一个方向使目标函数的值有所下降。从中选择一个下降方向 \mathbf{P}_k ，再沿着 \mathbf{P}_k 方向适当迈进一步，即在直线

$$\mathbf{X} = \mathbf{X}_k + \lambda \mathbf{P}_k \quad (1.9.5)$$

上，适当地确定一个新点

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \lambda_k \mathbf{P}_k \quad (1.9.6)$$

使得 $f(\mathbf{X}_{k+1}) < f(\mathbf{X}_k)$ 。

我们称 \mathbf{P}_k 为函数在 \mathbf{X}_k 处的下降方向（也称为搜索方向）， λ_k 称为步长因子。如何确定搜索方向 \mathbf{P}_k 和步长因子 λ_k ，这正是最优化方法应解决的两个问题。

一旦 \mathbf{P}_k 已经确定，如果要求 λ_k 使

$$f(\mathbf{X}_k + \lambda_k \mathbf{P}_k) = \min f(\mathbf{X}_k + \lambda \mathbf{P}_k) \quad (1.9.7)$$

成立，即通过 \mathbf{X}_k 点沿 \mathbf{P}_k 方向，求目标函数的极小点来求得 λ_k 。按式 (1.9.7) 求解 λ_k 是第 2 章一维搜索（或直线搜索）要解决的问题。

定理 1.9.1 若目标函数 $f(\mathbf{X})$ 具有连续的偏导数，如果 $\mathbf{X}_{k+1} = \mathbf{X}_k + \lambda_k \mathbf{P}_k$ 是函数 $f(\mathbf{X}_k + \lambda \mathbf{P}_k)$ 的极小点，则

$$\nabla f(\mathbf{X}_{k+1})^T \mathbf{P}_k = 0 \quad (1.9.8)$$

成立。

证 设 $\varphi(\lambda) = f(\mathbf{X}_k + \lambda \mathbf{P}_k)$ ，得

$$\varphi'(\lambda_k) = \nabla f(\mathbf{X}_k + \lambda_k \mathbf{P}_k)^T \mathbf{P}_k = \nabla f(\mathbf{X}_{k+1})^T \mathbf{P}_k = 0$$

证毕。

现将下降迭代法归纳如下：

(1) 选定初始点 \mathbf{X}_0 ，置 $k=0$ ，一般应尽可能使 \mathbf{X}_0 靠近函数 $f(\mathbf{X})$ 的极小点（最优点） \mathbf{X}^* 。

(2) 设已经求得 \mathbf{X}_k ，且 \mathbf{X}_k 不是最优点，则按某种算法的规则确定一个下降方向 \mathbf{P}_k ，即沿 \mathbf{P}_k 方向目标函数 $f(\mathbf{X})$ 的值有所下降。

(3) 按某种算法的规则确定 λ_k ，使

$$f(\mathbf{X}_k + \lambda_k \mathbf{P}_k) < f(\mathbf{X}_k)$$

成立。

(4) 计算 $\mathbf{X}_{k+1} = \mathbf{X}_k + \lambda_k \mathbf{P}_k$ 。

(5) 验证 \mathbf{X}_{k+1} 是否满足收敛准则，如果满足收敛准则，置 $\mathbf{X}^* = \mathbf{X}_{k+1}$ ，并打印出最优解；否则置 $k=k+1$ ，转到步骤 (2)。

1.9.2 收敛速度与收敛准则

作为一个算法，迭代序列 $\{X_k\}$ 能够收敛于问题的解 X^* 当然是必要的，但是，刚能收敛还不够，还必须以较快的速度收敛于问题的解。收敛的快慢用收敛的阶来度量。

定义 1.9.1 对收敛于解 X^* 的序列 $\{X_k\}$ ，若存在一个与 k 无关的数 $\beta \in [0,1]$ ，当 k 大于某一个 k_0 ，总有

$$\|X_{k+1} - X^*\| \leq \beta \|X_k - X^*\| \quad (1.9.9)$$

成立，则称序列 $\{X_k\}$ 为一阶收敛（或线性收敛）。

定义 1.9.2 对收敛于解 X^* 的序列 $\{X_k\}$ ，若存在一个与 k 无关的数 $\beta > 0$ ，当 k 大于某一个 k_0 ，总有

$$\|X_{k+1} - X^*\| \leq \beta \|X_k - X^*\|^\alpha \quad (1.9.10)$$

成立，则称序列 $\{X_k\}$ 为 α 阶收敛。

当 $\alpha=2$ 时，称为二阶收敛；当 $0 < \alpha < 2$ ，称为超线性收敛。

我们经常用到的收敛准则有：

(1) 两个相邻的迭代点的范数小于预先给定的值 ε ，即

$$\|X_{k+1} - X_k\| < \varepsilon \quad (1.9.11)$$

或其相对误差小于预先给定的值 ε ，即

$$\frac{\|X_{k+1} - X_k\|}{\|X_k\|} < \varepsilon \quad (1.9.12)$$

(2) 两个相邻迭代点的函数值的误差绝对值小于预先给定的值 ε ，即

$$|f(X_{k+1}) - f(X_k)| < \varepsilon \quad (1.9.13)$$

或其相对误差的绝对值小于预先给定的值 ε ，即

$$\left| \frac{f(X_{k+1}) - f(X_k)}{f(X_k)} \right| < \varepsilon \quad (1.9.14)$$

(3) X_k 点处函数的梯度的范数小于预先给定的值 ε ，即

$$|\nabla f(X_k)| < \varepsilon \quad (1.9.15)$$

如果算法利用了目标函数的梯度，一般用收敛准则 (3)；否则，用收敛准则 (1) 或收敛准则 (2) 作为算法的终止准则。

第2章 一维搜索

本章讨论的问题是仅含一个自变量的函数的极小点的求解问题，即求

$$\min f(x) \quad (2.0.1)$$

其中， $f: \mathbf{R}^1 \rightarrow \mathbf{R}^1$ ， $x \in \mathbf{R}^1$ 。求解问题 (2.0.1) 的迭代方法称为一维搜索或直线搜索。一维搜索不仅对于解决一维最优化问题具有实际意义，而且，在后面叙述的多维最优化问题的许多算法中，要求有一种有效的一维搜索方法。

求解一维问题的最优解的方法很多，本章讨论在多维最优化问题中经常用到的几种一维搜索方法。

1. 平分法 $\begin{cases} \text{利用函数 } f(x) \text{ 的一阶导数平分法} \\ \text{仅利用函数 } f(x) \text{ 本身的平分法} \end{cases}$

2. 黄金分割法或 0.618 法

3. 抛物线插值法

大部分一维搜索方法的求解过程可分为如下两个阶段：

(1) 确定极小点 x^* 所在的初始区间；

(2) 逐步缩小极小点 x^* 所在的初始区间，直到满足精度为止。

2.1 搜索区间的确定

2.1.1 单谷函数与搜索区间

在本章的讨论中，假设函数为单谷函数。下面，叙述单谷函数的定义。

定义 2.1.1 设 x^* 为函数 $f(x)$ 在区间 $[a, b]$ 内的全局极小点，如果在区间 $[a, b]$ 任取两个点 x_1 和 x_2 ，且 $x_1 < x_2$ ，都有

$$f(x_1) > f(x_2) \quad x_2 \leq x^* \quad (2.1.1)$$

$$f(x_1) < f(x_2) \quad x_1 > x^* \quad (2.1.2)$$

则称函数 $f(x)$ 为区间 $[a, b]$ 内的单谷函数。

单谷函数的例子如图 2.1.1 所示。

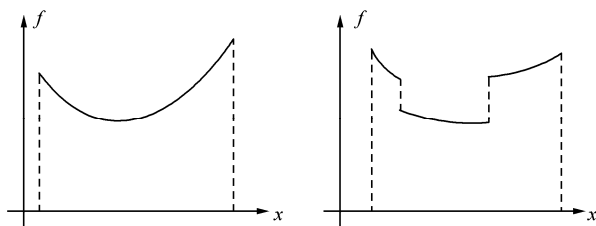


图 2.1.1 单谷函数

定义 2.1.2 $f: [a, b] \rightarrow \mathbf{R}^1$, x^* 为函数 $f(x)$ 在 $[a, b]$ 上的全局极小点。如果存在一个 x_1 和 x_2 , 且 $a \leq x_1$, $x_2 \leq b$, 使得

$$x_1 < x^* < x_2 \quad (2.1.3)$$

则称闭区间 $[x_1, x_2]$ 为函数 $f(x)$ 的极小点的一个搜索区间。

单谷函数有一个很有用的性质。利用这个性质, 可求得极小点的一个范围小的新的搜索区间, 从而可以无限地缩小搜索区间, 直到求得函数 $f(x)$ 的极小点。

单谷函数的性质 设 $[a, b]$ 为单谷函数 $f(x)$ 的极小点的一个搜索区间, 在 $[a, b]$ 内任取 x_1 和 x_2 , 且 $x_1 < x_2$ 。若 $f(x_1) < f(x_2)$, 则 $[a, x_2]$ 为 $f(x)$ 的极小点的一个搜索区间; 若 $f(x_1) > f(x_2)$, 则 $[x_1, b]$ 为 $f(x)$ 的极小点的一个搜索区间; 若 $f(x_1) = f(x_2)$, 则 $[a, x_2]$ 、 $[x_1, b]$ 和 $[x_1, x_2]$ 皆为 $f(x)$ 的极小点的一个搜索区间。

证 用反证法来证明, 分三种情况来证明。

(1) 若 $f(x_1) < f(x_2)$, 假设 $[a, x_2]$ 不是 $f(x)$ 的极小点的一个搜索区间, 那么, $f(x)$ 的极小点必在 $[x_2, b]$ 中。有 $x_2 \leq x^*$, 根据单谷函数的定义, 有 $f(x_1) > f(x_2)$, 这与 $f(x_1) < f(x_2)$ 的条件相矛盾, 故 $[a, x_2]$ 为 $f(x)$ 的极小点的一个搜索区间。

(2) 同理可证, 若 $f(x_1) > f(x_2)$, 则 $[x_1, b]$ 为 $f(x)$ 的极小点的一个搜索区间。

(3) 若 $f(x_1) = f(x_2)$, 假设 $[x_1, x_2]$ 不是 $f(x)$ 的极小点的一个搜索区间, 那么 $f(x)$ 的极小点 x^* 在区间 $[a, x_1]$ 或 $[x_2, b]$ 中, 不妨假设 $x^* \in [a, x_1]$, 根据单谷函数的定义, 有 $f(x_1) < f(x_2)$, 这与 $f(x_1) = f(x_2)$ 相矛盾, 故 $[x_1, x_2]$ 是 $f(x)$ 的极小点的一个搜索区间。当然, $[a, x_2]$ 和 $[x_1, b]$ 亦是 $f(x)$ 的极小点的一个搜索区间, 证毕。

对于单谷函数 $f(x)$ 来说, 如果 $x_1 < x_2 < x_3$ (或 $x_3 < x_2 < x_1$) 使

$$f(x_1) \geq f(x_2) \leq f(x_3) \quad (2.1.4)$$

成立。根据单谷函数的性质可知, 区间 $[x_1, x_3]$ (或区间 $[x_3, x_1]$) 是单谷函数 $f(x)$ 的极小点的一个搜索区间。

2.1.2 确定搜索区间的进退法

确定搜索区间 $[a, b]$ 有两种方法, 一种是仅利用函数本身的进退法, 另一种是利用函数的导数的进退法。

首先, 介绍仅利用函数本身来确定函数极小点的一个搜索区间 $[a, b]$ 的进退法, 其主要步骤为:

(1) 给定初始点 x_0 和步长 h , $h > 0$ 。

(2) 计算 $f(x_0)$ 和 $f(x_0+h)$ 。

(3) 比较 $f(x_0)$ 和 $f(x_0+h)$ ，有如下三种情况：

① 若 $f(x_0)=f(x_0+h)$ ，则 $[x_0, x_0+h]$ 为搜索区间；

② 若 $f(x_0)>f(x_0+h)$ ，说明 x^* 在 x_0 的右边，转到步骤 (4)；

③ 若 $f(x_0)<f(x_0+h)$ ，说明 x^* 在 x_0+h 的左边，置 $x_0=x_0+h$ ， $h=-h$ ，转到步骤 (4)。

(4) 计算 $f[x_0+(2^k-1)h]$ ($k=1,2,\dots$)，直到某一个 m ，使得

$$f[x_0+(2^{m-1}-1)h] \geq f[x_0+(2^m-1)h] \leq f[x_0+(2^{m+1}-1)h]$$

成立。令 $x_1=x_0+(2^{m-1}-1)h$ ， $x_2=x_0+(2^{m+1}-1)h$ 。

(5) 令 $a=\min\{x_1, x_2\}$ ， $b=\max\{x_1, x_2\}$ ，则区间 $[a, b]$ 为单谷函数 $f(x)$ 的极小点的一个搜索区间。

利用函数本身来确定函数的极小点的一个搜索区间 $[a, b]$ 的进退法程序框图如图 2.1.2 所示。

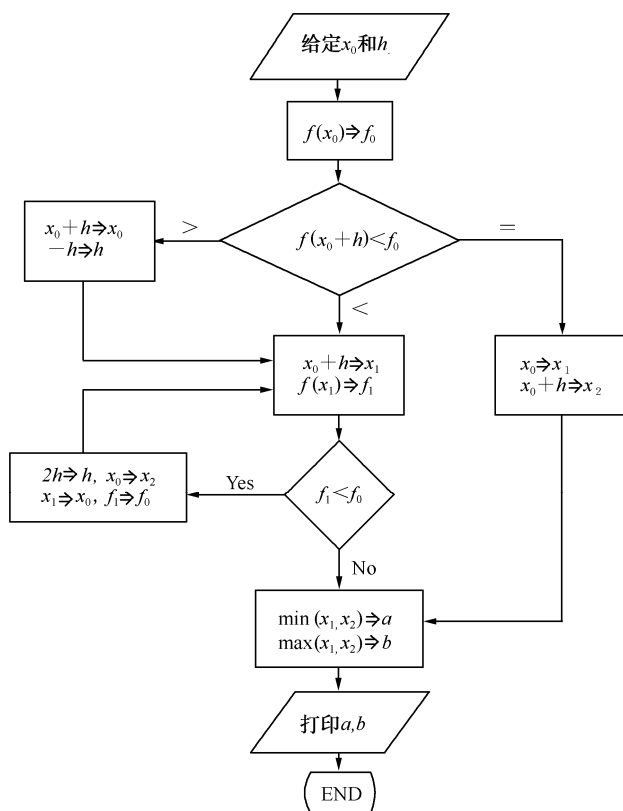


图 2.1.2 利用函数本身的进退法程序框图

例 2.1.1 求函数 $f(x)=(100-x)^2$ 的极小点 x^* (显然 $x^*=100$) 的一个搜索区间，取初始点 $x_0=30$ ， $h=5$ 。

解 计算并比较 $f(x_0)$ 和 $f(x_0+h)$ ：

$$f(x_0)=f(30)=4900$$

$$f(x_0+h)=f(35)=4225$$

由于 $f(x_0) > f(x_0 + h)$, 按算法做第 4 步。当 $m=4$, 有

$$\begin{aligned} f[x_0 + (2^{m-1} - 1)h] &= f(65) = 1225 \\ f[x_0 + (2^m - 1)h] &= f(105) = 25 \\ f[x_0 + (2^{m+1} - 1)h] &= f(185) = 7225 \end{aligned}$$

比较得

$$f(65) \geq f(105) \leq f(185)$$

所以, 区间 $[65, 185]$ 是函数 $f(x) = (100 - x)^2$ 的极小点 x^* 的一个搜索区间。

步长 h 的选定直接影响算法的收敛速度。如果选得过小, 需要迭代许多次才能找到搜索区间; 如果选取太大, 虽然一步就可能得到极小点的一个搜索区间, 但给下一步搜索极小点的过程增加了负担。文献[4]提出了如下的估算式:

$$h = \frac{2[f_e - f(x_0)]}{f'(x_0)} \quad (2.1.5)$$

其中, f_e 是 $f(x^*)$ 的估算值。如果 $f(x)$ 代表实际问题的误差函数, 可取 $f_e = 0$ 。

在求解多维最优化问题时, 如果采用下降迭代法, 而且, 每次迭代是使用一维搜索方法来确定下一个迭代点, 每次迭代都需要估计初始搜索步长 h 。

下面, 我们讨论在求解多维最优化问题中初始搜索步长 h 的确定。

现在, 考虑从 \mathbf{X}_k 到 \mathbf{X}_{k+1} 的迭代情况。设已经求得了迭代点 \mathbf{X}_k , 并按某种算法确定了搜索方向 \mathbf{P}_k , 下一个迭代点 \mathbf{X}_{k+1} 由如下一维搜索确定:

$$\begin{aligned} f(\mathbf{X}_k + \lambda_k \mathbf{P}_k) &= \min f(\mathbf{X}_k + \lambda \mathbf{P}_k) \\ \mathbf{X}_{k+1} &= \mathbf{X}_k + \lambda_k \mathbf{P}_k \end{aligned}$$

设 $\varphi(\lambda) = f(\mathbf{X}_k + \lambda \mathbf{P}_k)$, 比较并代入式 (2.1.5) 可得

$$h = \frac{2[f_e - f(\mathbf{X}_k)]}{\nabla f(\mathbf{X}_k)^T \mathbf{P}_k} \quad (2.1.6)$$

其中, f_e 是 $f(\mathbf{X}^*)$ 的估算值。

亦可采用下式来估计搜索步长 h :

$$h = \|\mathbf{X}_k - \mathbf{X}_{k-1}\| \quad (2.1.7)$$

当 $k=0$ 时, 可用式 (2.1.6) 来确定搜索步长 h , 亦可以预先给定搜索步长 h 。

确定单谷函数 $f(x)$ 的极小点的一个搜索区间的另一种方法是利用函数的导数来确定。下面, 我们来讨论利用函数的导数确定函数极小点的一个搜索区间的进退法。

对于具有一阶连续导数的单谷函数来说, 有 $f(x^*) = 0$ 。当 $x < x^*$ 时, 有 $f'(x) < 0$; 当 $x > x^*$ 时, 有 $f'(x) > 0$ 。因此, $[a, b]$ 为单谷函数的一个搜索区间必有 $f'(a) < 0$ 和 $f'(b) > 0$ 。

利用函数的导数来确定函数极小点的一个搜索区间 $[a, b]$ 的主要步骤如下:

(1) 给定初始点 x_0 和步长 h , $h > 0$ 。

(2) 计算 $f'(x_0)$ 。

(3) 比较 $f'(x_0) = 0$, 则 x_0 为极小点。

(4) 若 $f'(x_0) < 0$, 计算 $f'[x_0 + (2^k - 1)h]$ ($k=1, 2, \dots$), 直到某一个 m , 使得

$$f'[x_0 + (2^m - 1)h] > 0$$

成立。 $a=x_0+(2^{m-1}-1)h$, $b=x_0+(2^m-1)h$ 。

(5) 若 $f'(x_0) > 0$, 计算 $f'[x_0-(2^k-1)h]$ ($k=1,2,\dots$), 直到某一个 m , 使得

$$f'[x_0-(2^m-1)h] < 0$$

成立。 $a=x_0-(2^{m-1}-1)h$, $b=x_0-(2^m-1)h$ 。

则区间 $[a,b]$ 为单谷函数 $f(x)$ 的极小点的一个搜索区间。

例 2.1.2 利用函数本身的进退法求函数 $f(x)=(100-x)^2$ 的极小点的一个搜索区间, 取初始点 $x_0=30$, $h=5$ 。

解

$$f'(x)=2(x-100)$$

$$f'(30)=-140$$

由于 $f'(x_0) < 0$, 按算法做第 4 步。当 $m=4$, 有

$$f'[x_0+(2^m-1)h]=f'(105)=10 > 0$$

所以, $a=x_0+(2^{m-1}-1)h=65$, $b=x_0+(2^m-1)h=105$, 故区间 $[65,105]$ 是函数 $f(x)=(100-x)^2$ 的极小点 x^* 的一个搜索区间。

利用函数的导数确定函数的极小点的一个搜索区间 $[a,b]$ 的进退法程序框图如图 2.1.3 所示。

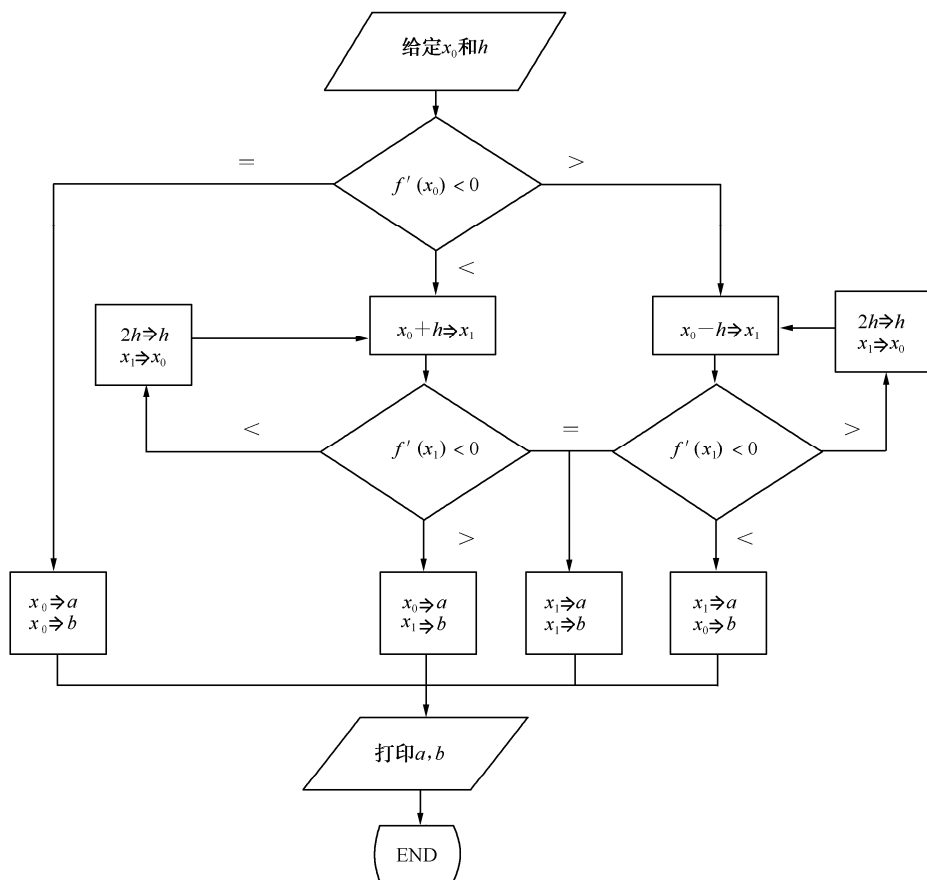


图 2.1.3 利用函数导数的进退法程序框图

上述两种确定单谷函数极小点的一个搜索区间的算法可供选用。如果容易求得函数的导数, 选用利用函数导数的进退法要好一些。但是, 如果函数的导数难以求得, 或函数的导数不连续时, 这时, 只能选用利用函数本身的进退法。

2.2 平分法

平分法分为利用函数导数的平分法和仅利用函数本身的平分法两种方法, 现分别叙述如下。

2.2.1 利用函数导数的平分法

设 $f(x)$ 的极小点的一个搜索区间 $[a, b]$ 已经由上节讨论的进退法确定。由上节的讨论可知, $f'(a) < 0$ 和 $f'(b) > 0$ 。计算搜索区间的中点 $x = \frac{a+b}{2}$ 和 $f'(x)$, 如图 2.2.1 所示。如果 $f'(x) < 0$, 则 $[x, b]$ 为新的搜索区间; 如果 $f'(x) > 0$, 则 $[a, x]$ 为新的搜索区间; $f'(x) = 0$, 则 x 为函数的极小点。再计算新搜索区间的中点, 利用函数的导数确定下一个新搜索区间。反复计算搜索区间中点的导数和确定新的搜索区间, 直到搜索区间的长度小于预先给定的精度值为止。

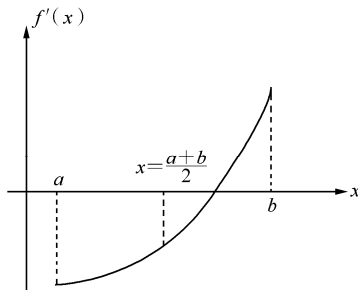


图 2.2.1 利用函数导数的平分法

利用函数导数的平分法的主要步骤如下:

- (1) 给定初始点 x_0 、步长 h 和收敛精度 ε 。
 - (2) 用进退法确定函数 $f(x)$ 的极小点的一个搜索区间 $[a, b]$ 。
 - (3) 计算 $x = \frac{a+b}{2}$ 和 $f'(x)$ 。
 - (4) 若 $f'(x) < 0$, 置 $a = x$, 转到步骤 (5); 若 $f'(x) = 0$, 置 $x^* = x$, 转到步骤 (6); 若 $f'(x) > 0$, 置 $b = x$, 转到步骤 (5)。
 - (5) 若 $|b - a| < \varepsilon$, 置 $x^* = \frac{a+b}{2}$, 转到步骤 (6); 否则, 转到步骤 (3)。
 - (6) 打印 x^* 和 $f(x^*)$, 停止程序执行。
- 利用函数导数的平分法程序框图如图 2.2.2 所示。

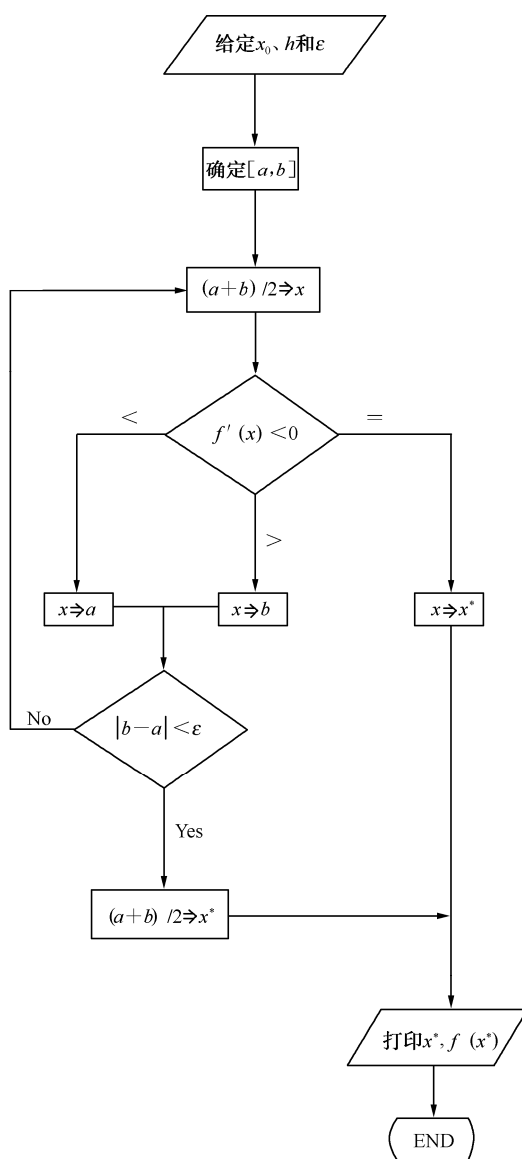


图 2.2.2 利用函数导数的平分法程序框图

例 2.2.1 求函数 $f(x) = (100 - x)^2$ 的极小点, 取初始点 $x_0 = 30$, $h = 5$ 。

初始搜索区间 $[a, b]$ 在例 2.1.2 中已经求得, 利用函数导数的平分法求解 $\min f(x) = \min(100 - x)^2$ 的极小点有关数据如表 2.2.1 所示。

表 2.2.1 利用函数导数的平分法计算的有关数据

| 迭代次数 | a | b | $\frac{a+b}{2}$ | $f'(x)$ |
|------|---------------------|-----|-----------------|---------|
| 1 | 65 | 105 | 85 | -30 |
| 2 | 85 | 105 | 95 | -10 |
| 3 | 95 | 105 | 100 | 0 |
| 最优解 | $x^*=100, f(x^*)=0$ | | | |

2.2.2 仅利用函数本身的平分法

设 $f(x)$ 的极小点的一个搜索区间 $[a, b]$ 已经由上节的进退法确定。将搜索区间 $[a, b]$ 分为四等分, 如图 2.2.3 所示, 设等分点分别为 x_1 、 x_m 和 x_2 , 且 $x_1 < x_m < x_2$, 对应的函数值分别为 f_1 、 f_m 和 f_2 。然后, 比较等分点函数值的大小, 根据单谷函数的性质确定下一次迭代新的搜索区间。再用相同的方法进行等分、比较确定再下一次迭代新的搜索区间。如此反复地迭代, 直到搜索区间的长度 $|b-a| < \varepsilon$ 为止, ε 为迭代收敛精度。

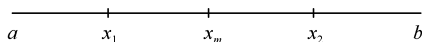


图 2.2.3 利用函数本身的平分法

仅利用函数本身的平分法的主要步骤如下:

- (1) 给定初始点 x_0 、步长 h 和收敛精度 ε 。
 - (2) 用进退法确定函数 $f(x)$ 的极小点的一个搜索区间 $[a, b]$ 。
 - (3) 计算 $x_m = \frac{a+b}{2}$ 、 $f(x_m)$ 和 $L = b - a$ 。
 - (4) 计算 $x_1 = a + \frac{L}{4}$ 、 $x_2 = b - \frac{L}{4}$ 、 $f(x_1)$ 和 $f(x_2)$ 。
 - (5) 比较 $f(x_1)$ 和 $f(x_m)$, 若 $f(x_1) < f(x_m)$, 置 $b = x_m$, $x_m = x_1$, 转到步骤 (7); 若 $f(x_1) \geq f(x_m)$, 转到步骤 (6)。
 - (6) 比较 $f(x_2)$ 和 $f(x_m)$, 若 $f(x_2) < f(x_m)$, 置 $a = x_m$, $x_m = x_2$; 若 $f(x_2) \geq f(x_m)$, 置 $a = x_1$, $b = x_2$, x_m 仍是新搜索区间的中点。
 - (7) 计算 $L = b - a$ 。
 - (8) 若 $L < \varepsilon$, 则置 $x^* = \frac{a+b}{2}$, 转到步骤 (9); 否则, 转到步骤 (4)。
 - (9) 打印 x^* 和 $f(x^*)$, 停止程序执行。
- 利用函数本身的平分法程序框图如图 2.2.4 所示。

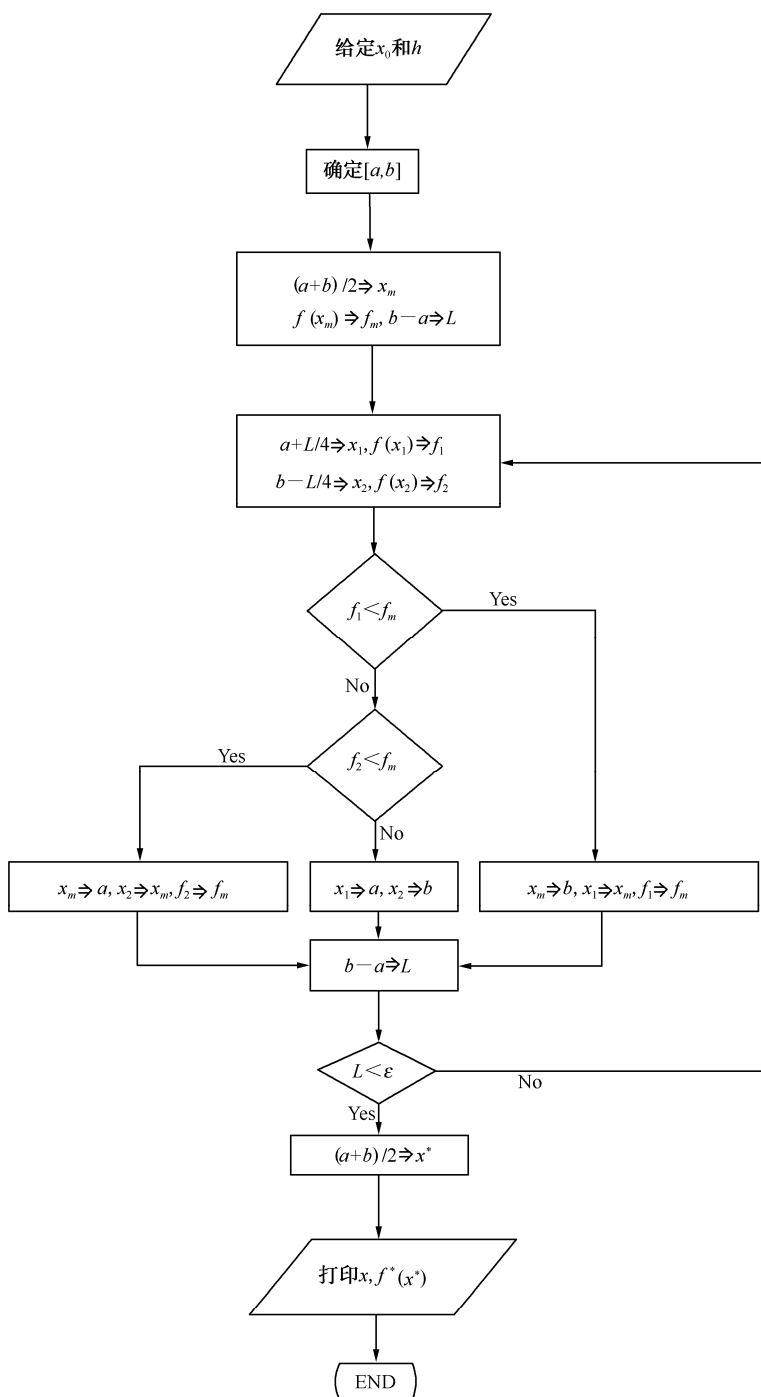


图 2.2.4 利用函数本身的平分法程序框图

例 2.2.2 求函数 $f(x)=(100-x)^2$ 的极小点, 取初始点 $x_0=30$, $h=5$ 。

初始搜索区间 $[a, b]$ 在例 2.1.1 中已经求得, 利用函数本身的平分法计算的数据如表 2.2.2 所示。

表 2.2.2 利用函数本身的平分法计算的有关数据

| 迭代次数 | a | b | x_1 | x_m | x_2 |
|------|------------------------|---------|----------|----------|----------|
| 0 | 65 | 185 | 95 | 125 | 155 |
| 1 | 65 | 125 | 80 | 95 | 110 |
| 2 | 80 | 110 | 87.5 | 95 | 102.5 |
| 3 | 95 | 110 | 98.75 | 102.5 | 106.25 |
| 4 | 95 | 102.5 | 96.875 | 98.75 | 100.625 |
| 5 | 98.75 | 102.5 | 99.6875 | 100.625 | 101.5625 |
| 6 | 98.75 | 100.625 | 99.21875 | 99.6875 | 100.1563 |
| ... | ... | ... | ... | ... | ... |
| 20 | 99.99992 | 100 | 99.99996 | 99.99999 | 100 |
| 21 | 99.99999 | 100 | 100 | 100 | 100 |
| 最优解 | $x^*=100$, $f(x^*)=0$ | | | | |

上述两种方法经过一次迭代, 搜索区间正好缩小了一半, 故称它们为平分法。方法一用了函数的导数, 对于容易求得导数的函数, 方法一的计算量比方法二的计算量少。方法二的优点是不涉及函数的导数, 但从计算量来衡量, 它不如下面叙述的黄金分割法, 因此, 一般很少用方法二。

2.3 黄金分割法

从上节中, 我们知道在函数 $f(x)$ 的极小点 x^* 的一个搜索区间内插入三个点, 比较它们的函数值的大小就能缩小搜索区间。由单谷函数的性质可知, 在搜索区间内插入两个点, 亦可将搜索区间缩小。现在, 有这样一个问题, 在计算函数值次数相同的条件下, 如何使搜索区间缩小最快。换句话说, 用最小的计算函数值的次数, 将搜索区间缩小到满足收敛精度要求。

从理论上可以证明黄金分割法是在计算函数值次数相同的条件下, 搜索区间缩小最快的方法。理论上的证明请参看有关资料, 下面, 我们仅做直观分析。

首先, 在搜索区间内对称地放置两个插入点 x_1 和 x_2 (满足 $a < x_1 < x_2 < b$), 使 $x_1 - a = b - x_2$, 这样, 无论删掉哪一段, 保留的搜索区间的长度是相同的, 如图 2.3.1 所示。

令 $\alpha = \frac{x_2 - a}{b - a} > 0$, 可以推出插入点 x_1 和 x_2 的公式为

$$\begin{cases} x_1 = b - \alpha(b - a) \\ x_2 = a + \alpha(b - a) \end{cases} \quad (2.3.1)$$

通过插入点的函数值比较, 得到新的搜索区间, 我们不妨假设新的搜索区间为 $[a, x_2]$ 。

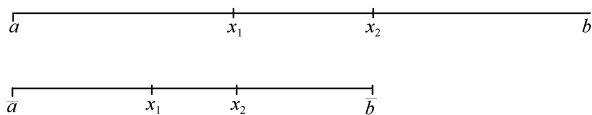


图 2.3.1 黄金分割法

其次, 为了减少计算函数值的次数, 我们希望新的搜索区间内的 x_1 点能作为第二次迭代按式 (2.3.1) 计算的新的搜索区间的 \bar{x}_2 点。为了与第一次迭代区分开来, 用 $[\bar{a}, \bar{b}]$ 表示第二次迭代的搜索区间, 用 \bar{x}_1 和 \bar{x}_2 表示第二次迭代搜索区间内的两个插入点。根据假设, 新的搜索区间为 $[a, x_2]$, 有

$$\begin{aligned}\bar{a} &= a, \quad \bar{b} = x_2 \\ \bar{x}_2 &= \bar{a} + \alpha(\bar{b} - \bar{a}) = a + (x_2 - a) = a + \alpha^2(b - a)\end{aligned}$$

要求 $\bar{x}_2 = x_1$, 所以有

$$a + \alpha^2(b - a) = b - \alpha(b - a)$$

由于 $b - a > 0$, 由上式可得

$$\alpha^2 + \alpha - 1 = 0$$

解上式得

$$\alpha = \frac{-1 + \sqrt{5}}{2} = 0.61803399$$

在古代, 人们认为按上式 α 的比率分割线段是最协调的, 胜似黄金, 故将按此比率插值进行的一维搜索方法称为黄金分割法。由于 $\alpha \approx 0.618$, 故又称此算法为 0.618 法。

黄金分割法的主要步骤:

(1) 给定初始点 x_0 、步长 h 和收敛精度 ε 。

(2) 用进退法确定函数 $f(x)$ 的极小点的一个搜索区间 $[a, b]$ 。

(3) 计算 $\begin{cases} x_1 = b - \alpha(b - a) \\ x_2 = a + \alpha(b - a) \end{cases}$, 其中 $\alpha = \frac{-1 + \sqrt{5}}{2}$ 。

(4) 比较 $f(x_1)$ 和 $f(x_2)$ 。

(5) 若 $f(x_1) \leq f(x_2)$, 置 $b = x_2$, $x_2 = x_1$, 求新搜索区间的插入点 $x_1 = b - \alpha(b - a)$, 转到步骤 (6); 若 $f(x_1) > f(x_2)$, 置 $a = x_1$, $x_1 = x_2$, 求新搜索区间的插入点 $x_2 = a + \alpha(b - a)$ 。

(6) 若 $b - a < \varepsilon$, 置 $x^* = \frac{a+b}{2}$, 转到步骤 (7); 否则, 转到步骤 (4)。

(7) 打印 x^* 和 $f(x^*)$, 停止程序执行。

黄金分割法程序框图如图 2.3.2 所示。

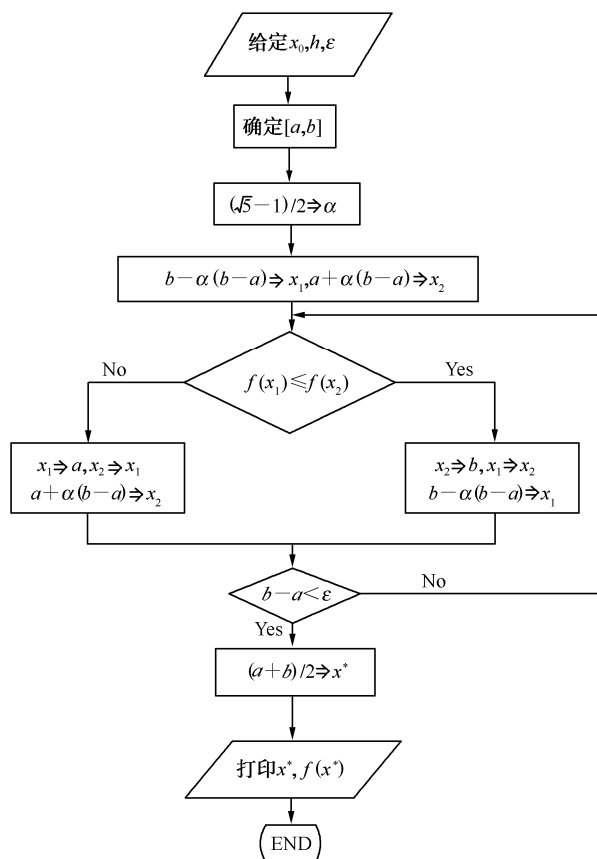


图 2.3.2 黄金分割法程序框图

例 2.3.1 求函数 $f(x) = (100 - x)^2$ 的极小点，取初始点 $x_0 = 30$ ， $h = 5$ 。

初始搜索区间 $[a, b]$ 在例 2.1.1 中已经求得，利用黄金分割法计算的数据如表 2.3.1 所示。

表 2.3.1 利用黄金分割法计算的有关数据

| 迭代次数 | a | b | x_1 | x_2 |
|------|----------------------------|----------|----------|----------|
| 0 | 65 | 185 | 110.8359 | 139.1641 |
| 1 | 65 | 139.1641 | 93.32816 | 110.8359 |
| 2 | 65 | 110.8359 | 82.50777 | 93.32816 |
| 3 | 82.50777 | 110.8359 | 93.32816 | 100.0155 |
| 4 | 93.32816 | 110.8359 | 100.0155 | 104.1486 |
| 5 | 93.32816 | 104.1486 | 97.46118 | 100.0155 |
| 6 | 97.46118 | 104.1486 | 100.0155 | 101.5942 |
| ... | ... | ... | ... | ... |
| 30 | 99.99998 | 100 | 100 | 100 |
| 最优解 | $x^* = 100$ ， $f(x^*) = 0$ | | | |

黄金分割法的优点是仅利用函数本身, 不涉及函数的导数; 只要函数 $f(x)$ 是单谷函数, 总能求得其极小点 x^* , 收敛速度较快。

2.4 抛物线插值法

抛物线插值法也是一种迭代方法, 它通过函数 $f(x)$ 极小点的搜索区间内三个点的函数值作二次抛物线插值, 然后, 求抛物线的极小点来近似地代替函数的极小点, 用所求得的极小点代替原三个点中的一个点。为了使抛物线的极值点为极小点, 要求搜索区间内三个插值点满足如下关系: 当 $x_1 < x_2 < x_3$ 时, 有 $f(x_1) \geq f(x_2) < f(x_3)$ 或 $f(x_1) > f(x_2) \leq f(x_3)$ 成立。

首先, 推导插值抛物线的极小点 x_4 的求解公式。对于给定的三个不同点 x_1 、 x_2 和 x_3 , 以及它们相应的函数值 f_1 、 f_2 和 f_3 。设插值抛物线方程为

$$q(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2) \quad (2.4.1)$$

其中, 三个待定常数 a_0 、 a_1 和 a_2 由插值点的函数值来确定。

当 $x = x_1$ 时, 有

$$f_1 = q(x_1) = a_0$$

故

$$a_0 = f_1 \quad (2.4.2)$$

当 $x = x_2$ 时, 有

$$f_2 = q(x_2) = a_0 + a_1(x_2 - x_1)$$

故

$$a_1 = \frac{f_2 - f_1}{x_2 - x_1} \quad (2.4.3)$$

当 $x = x_3$ 时, 有

$$f_3 = q(x_3) = a_0 + a_1(x_3 - x_1) + a_2(x_3 - x_1)(x_3 - x_2)$$

故

$$a_2 = \frac{1}{x_3 - x_2} \left(\frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1} \right) \quad (2.4.4)$$

对于给定的三个不同的点及它们的函数值, 按上述公式计算 a_0 、 a_1 和 a_2 , 就建立了抛物线插值公式。

求插值抛物线 (2.4.1) 的极值点, 令

$$q'(x) = a_1 + a_2(x - x_1) + a_2(x - x_2) = 0$$

解之得极值点

$$x_4 = \frac{x_1 + x_2}{2} - \frac{a_1}{2a_2} \quad (2.4.5)$$

其次, 我们来证明当三个插值点 $x_1 < x_2 < x_3$, 使得 $f(x_1) \geq f(x_2) < f(x_3)$ 或 $f(x_1) > f(x_2) \leq f(x_3)$ 成立时, 由式 (2.4.5) 求得的 x_4 是式 (2.4.1) 二次插值函数的极小点, 且有

$$\frac{x_1+x_2}{2} \leq x_4 \leq \frac{x_2+x_3}{2} \quad (2.4.6)$$

成立。

证 由于式 (2.4.5) 求得的 x_4 是式 (2.4.1) 二次函数的极值点, 要证明式 (2.4.5) 求得的 x_4 是式 (2.4.1) 二次函数的极小点, 只要证明 $a_2 > 0$ 就可以了。

令
由条件得

$$\begin{aligned} f &= \min \{f_1, f_3\} \\ f &\geq f_2 \end{aligned}$$

$$\begin{aligned} a_2 &= \frac{1}{x_3 - x_2} \left(\frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1} \right) \\ &= \frac{f_1(x_3 - x_2) + f_2(x_1 - x_3) + f_3(x_2 - x_1)}{(x_2 - x_1)(x_3 - x_2)(x_3 - x_1)} \\ &= \frac{f - f_2}{(x_2 - x_1)(x_3 - x_2)} + \frac{(f_1 - f)(x_3 - x_2) + (f_3 - f)(x_2 - x_1)}{(x_2 - x_1)(x_3 - x_2)(x_3 - x_1)} \\ &> \frac{f - f_2}{(x_2 - x_1)(x_3 - x_2)} \geq 0 \end{aligned}$$

所以 $a_2 > 0$, 故 x_4 为极小点。

由于 $x_1 < x_2 < x_3$, $f_1 \geq f_2$, 有 $a_1 < 0$ 。所以

$$x_4 = \frac{x_1+x_2}{2} - \frac{a_1}{2a_2} \geq \frac{x_1+x_2}{2}$$

交换 x_1 和 x_3 点, 同理可证

$$x_4 \leq \frac{x_2+x_3}{2}$$

证毕。

抛物线插值法的主要步骤如下:

- (1) 给定初始点 x_0 、步长 h 和收敛精度 ε 。
- (2) 用进退法确定 $x_1 < x_2 < x_3$, 使得 $f(x_1) \geq f(x_2) < f(x_3)$ 或 $f(x_1) > f(x_2) \leq f(x_3)$ 成立。
- (3) 按式 (2.4.5) 计算抛物线的极小点 x_4 。
- (4) 若 $|x_2 - x_4| < \varepsilon$, 转到步骤 (6)。
- (5) 保存最好点 x_2 或 x_4 (所谓最好点是指函数值最小的点) 和包含最好点的两个相邻点作为下一次迭代的三个插值点 x_1 、 x_2 和 x_3 , 转到步骤 (3)。
- (6) 以最好点作为最优点, 打印最优解, 停止程序执行。

抛物线插值法程序框图如图 2.4.1 所示。

抛物线插值法对于求解二次函数的极小点, 只需要一次迭代就能得到极小点 x^* 。一般非二次函数在极小点附近呈接近于抛物线, 因此, 抛物线插值法收敛速度快。但是, 即使对于单谷函数, 抛物线插值法也有可能收敛到一个非极小点上。

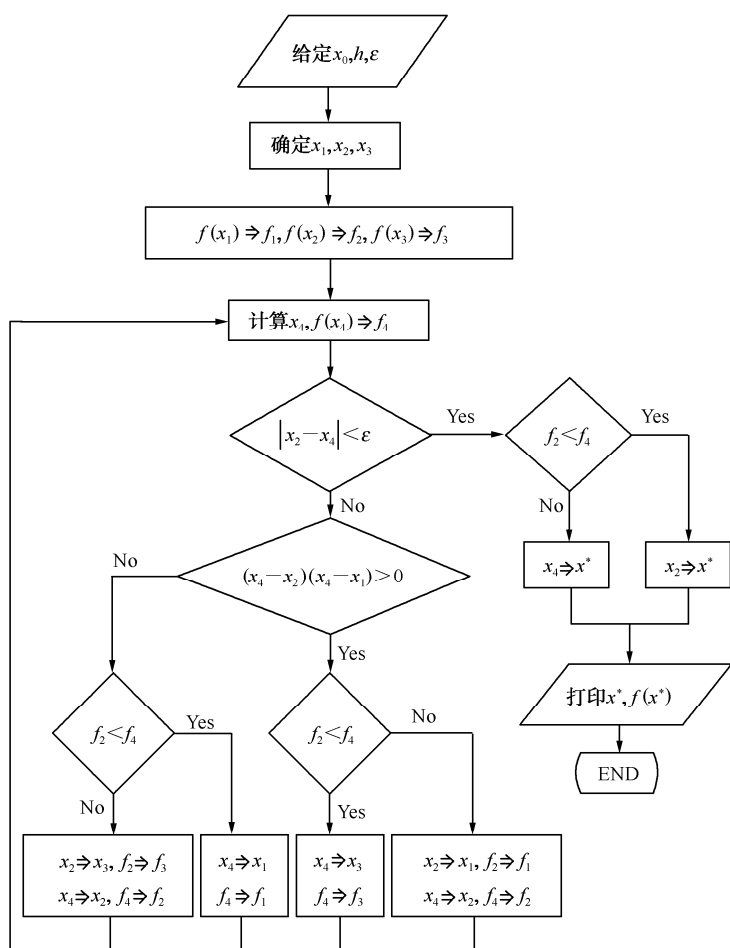


图 2.4.1 抛物线插值法程序框图

第3章 无约束最优化问题的求解方法

本章讨论多维无约束最优化问题，无约束最优化问题的一般形式为

$$\min f(X) \quad (3.0.1)$$

其中, $f: \mathbf{R}^n \rightarrow \mathbf{R}^1$ 。这个问题的求解是指在 \mathbf{R}^n 中找一点 X^* , 使得对于任意的 $X \in \mathbf{R}^n$, 都有

$$f(X^*) \leq f(X) \quad (3.0.2)$$

成立。 X^* 就是问题(3.0.1)的全局最优点。但是, 大多数最优化方法只能求到局部极小点。

无约束最优化方法可分为两大类, 一类是利用函数的导数的间接法, 这类无约束最优化方法有: 梯度法、共轭梯度法、牛顿法和变尺度法等; 另一类是仅利用函数本身的直接法, 这类无约束最优化方法有: 坐标轮换法、模式搜索法、Powell法和单纯形法等。

3.1 梯度法

梯度法是最古老、最基本的最优化方法之一。它具有算法简单、使用方便等优点; 同时, 一些更有效的最优化方法是对梯度法进行改进或在它的启发下而获得的。它的缺点是收敛速度慢。

有学者认为, 从工程实用上说, 梯度法是最好的方法之一, 其原因是梯度法程序简单和前面几步收敛速度快。下面, 我们来讨论梯度法。

设 X_0 为初始近似点, X_k 为已经求得的第 k 次迭代点。根据 1.9 节的下降迭代法, 第 $k+1$ 次迭代点为

$$X_{k+1} = X_k + \lambda P_k \quad (\lambda > 0)$$

将函数 $f(X_k + \lambda P_k)$ 进行 Taylor 展开, 有

$$f(X_k + \lambda P_k) = f(X_k) + \lambda \nabla f(X_k)^T P_k + \frac{1}{2} \lambda^2 \nabla^2 f(X_k) P_k P_k^T P_k + o(\lambda^2) \quad (3.1.1)$$

当 λ 很小时, $\frac{1}{2} \lambda^2 \nabla^2 f(X_k) P_k P_k^T P_k + o(\lambda^2)$ 相对于 λ 的一次项是高阶小量。因此, 当 $\lambda > 0$, 且很小时, 要使

$$f(X_k + \lambda P_k) < f(X_k) \quad (3.1.2)$$

成立, 必须

$$\nabla f(X_k)^T P_k < 0 \quad (3.1.3)$$

如果 $\nabla f(X_k) \neq 0$ 向量, 则满足式(3.1.3)的向量 P_k 有无穷多个。沿不同的搜索方向函数有不同的下降速度, 而负梯度方向是函数局部下降最快的方向。因此, 取搜索方向

$$P_k = -\nabla f(X_k) \quad (3.1.4)$$

步长 λ_k 的确定有三种方法:

(1) 取 $\lambda_k = c$, c 为预先给定的常数, 如果

$$f(\mathbf{X}_k + \lambda_k \mathbf{P}_k) < f(\mathbf{X}_k) \quad (3.1.5)$$

成立, 置 $\mathbf{X}_{k+1} = \mathbf{X}_k + \lambda_k \mathbf{P}_k$, 继续求下一个迭代点; 否则, 置 $\lambda_k = \alpha c$, $\alpha \in (0, 1)$, 再检验式 (3.1.5) 是否成立, 调整 α 的值, 直到式 (3.1.5) 成立。我们称这种方法为梯度法的可接受形式。由于它省去了一维求优的过程, 往往比最速下降法更为简单, 且更有效。

(2) 为了使函数在 \mathbf{P}_k 方向上的函数值下降最多, 沿 \mathbf{P}_k 进行一维搜索来确定 λ_k , 即使

$$\begin{cases} f(\mathbf{X}_k + \lambda_k \mathbf{P}_k) = \min_{\lambda} f(\mathbf{X}_k + \lambda \mathbf{P}_k) \\ \mathbf{X}_{k+1} = \mathbf{X}_k + \lambda_k \mathbf{P}_k \end{cases} \quad (3.1.6)$$

由式 (3.1.4) 和式 (3.1.6) 构成梯度法, 或称为最速下降法。

(3) 对于二次函数, λ_k 的确定有计算公式可循, 将梯度法应用于具有对称正定矩阵 \mathbf{A} 的二次函数

$$f(\mathbf{X}) = \frac{1}{2} \mathbf{X}^T \mathbf{A} \mathbf{X} + \mathbf{B}^T \mathbf{X} + c \quad (3.1.7)$$

可推出 λ_k 的显式迭代公式。

设 \mathbf{X}_k 为第 k 次迭代点, 则有

$$\mathbf{P}_k = -\nabla f(\mathbf{X}_k) = -\mathbf{A} \mathbf{X}_k - \mathbf{B} \quad (3.1.8)$$

令 $\varphi(\lambda) = f(\mathbf{X}_k + \lambda \mathbf{P}_k)$, 因为, $f(\mathbf{X}_k + \lambda_k \mathbf{P}_k) = \min_{\lambda} f(\mathbf{X}_k + \lambda \mathbf{P}_k)$, 有 $\varphi'(\lambda_k) = 0$, 由复合函数求导, 得

$$\varphi'(\lambda_k) = \nabla f(\mathbf{X}_k + \lambda_k \mathbf{P}_k)^T \mathbf{P}_k = 0 \quad (3.1.9)$$

而

$$\begin{aligned} \nabla f(\mathbf{X}_k + \lambda_k \mathbf{P}_k) &= \nabla f[\mathbf{X}_k - \lambda_k \nabla f(\mathbf{X}_k)] \\ &= \mathbf{A}[\mathbf{X}_k - \lambda_k \nabla f(\mathbf{X}_k)] + \mathbf{B} \\ &= \nabla f(\mathbf{X}_k) - \lambda_k \mathbf{A} \nabla f(\mathbf{X}_k) \end{aligned} \quad (3.1.10)$$

将式 (3.1.10) 和式 (3.1.8) 代入式 (3.1.9), 得

$$\lambda_k \nabla f(\mathbf{X}_k)^T \mathbf{A} \nabla f(\mathbf{X}_k) = \nabla f(\mathbf{X}_k)^T \nabla f(\mathbf{X}_k) \quad (3.1.11)$$

由于 \mathbf{A} 为正定矩阵, $\nabla f(\mathbf{X}_k) \neq 0$, 所以

$$\lambda_k = \frac{\nabla f(\mathbf{X}_k)^T \nabla f(\mathbf{X}_k)}{\nabla f(\mathbf{X}_k)^T \mathbf{A} \nabla f(\mathbf{X}_k)} \quad (3.1.12)$$

梯度法的主要步骤如下:

- (1) 给定初始点 \mathbf{X}_0 和收敛精度 ε 。
- (2) 设 \mathbf{X}_k 为第 k 次迭代点已经求得, 计算搜索方向 $\mathbf{P}_k = -\nabla f(\mathbf{X}_k)$ 。
- (3) 用一维搜索方法求 λ_k , 即使 $f(\mathbf{X}_k + \lambda_k \mathbf{P}_k) = \min_{\lambda} f(\mathbf{X}_k + \lambda \mathbf{P}_k)$, 置 $\mathbf{X}_{k+1} = \mathbf{X}_k + \lambda_k \mathbf{P}_k$ 。
- (4) 若 $\|\nabla f(\mathbf{X}_{k+1})\| < \varepsilon$, 置 $\mathbf{X}^* = \mathbf{X}_{k+1}$, 并打印最优解 \mathbf{X}^* 和 $f(\mathbf{X}^*)$, 停止程序执行; 否则, 置 $k = k+1$, $\mathbf{X}_k = \mathbf{X}_{k+1}$, 转到步骤 (2)。

梯度法程序的框图如图 3.1.1 所示。

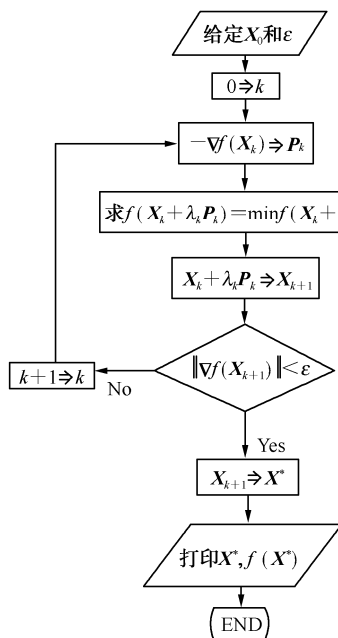


图 3.1.1 梯度法程序框图

在梯度法中，用一维搜索求搜索步长 λ_k ，即使 $f(X_k + \lambda_k P_k) = \min f(X_k + \lambda P_k)$ 。令 $\varphi(\lambda) = f(X_k + \lambda P_k)$ ，由复合函数求导，得

$$\nabla f(X_{k+1})^T P_k = 0$$

其中， $X_{k+1} = X_k + \lambda_k P_k$ 。因为，在梯度法中，取 $P_{k+1} = -\nabla f(X_{k+1})$ ，所以，有

$$P_{k+1}^T P_k = 0 \quad (3.1.13)$$

由上式知，在梯度法中，相邻的迭代搜索方向是相互垂直的，使搜索路线出现锯齿现象，在接近极小点附近齿形越来越小，从而导致梯度法收敛速度低。为了消除锯齿现象，我们取

$$X_{k+1} = X_k + \alpha \lambda_k P_k \quad (3.1.14)$$

一般 $\alpha = 0.8 \sim 0.99$ 。

例 3.1.1 求函数 $f(X) = x_1^2 + 25x_2^2$ 的极小点[精确解 $X^* = (0, 0)^T$]，取初始点 $X_0 = (70, 9)^T$ 。为了防止产生锯齿现象，取 $X_{k+1} = X_k + 0.98\lambda_k P_k$ 。用利用导数的平分法作为一维搜索方法，用作者自己编写的程序计算结果及有关计算数据如表 3.1.1 所示。

表 3.1.1 用梯度法程序计算的有关数据

| k | x_1^k | x_2^k | $\frac{\partial f(X_k)}{\partial x_1}$ | $\frac{\partial f(X_k)}{\partial x_2}$ | λ_k |
|-----|---------|---------|--|--|-------------|
| 0 | 70 | 9 | 140 | 450 | 0.0219 |
| 1 | 67.00 | -0.64 | 134.00 | -31.86 | 0.2189 |
| 2 | 38.26 | 6.20 | 76.51 | 309.88 | 0.0212 |
| 3 | 36.67 | -0.23 | 73.37 | -11.53 | 0.3166 |
| 4 | 13.91 | 3.35 | 27.88 | 167.39 | 0.0205 |

续表

| k | x_1^k | x_2^k | $\frac{\partial f(X_k)}{\partial x_1}$ | $\frac{\partial f(X_k)}{\partial x_2}$ | λ_k |
|-----|-----------------------|------------------------|--|--|-------------|
| 5 | 13.35 | -0.02 | 26.71 | -1.00 | 0.4840 |
| 6 | 0.70 | 0.46 | 1.39 | 22.80 | 0.0201 |
| 7 | 0.67 | 0.01 | 1.34 | 0.37 | 0.1834 |
| ... | ... | ... | ... | ... | ... |
| 24 | 1.65×10^{-5} | -7.80×10^{-5} | | | |

3.2 共轭梯度法

为了直观地了解共轭梯度法, 我们考虑二维二次函数

$$f(X) = \frac{1}{2} X^T A X + B^T X + c \quad (3.2.1)$$

其中, $X = \{x_1, x_2\}^T$, A 为 2×2 阶对称正定矩阵, $B = \{b_1, b_2\}^T$, c 为常数。取任意初始点 X_0 , 沿某个下降方向 [例如 $P_0 = -\nabla f(X_0)$ 方向] 作一维搜索得 X_1 , 如图 3.2.1 所示。因为 X_1 是通过 X_0 点沿 P_0 方向上函数的极小点, 故有

$$\nabla f(X_1)^T P_0 = 0 \quad (3.2.2)$$

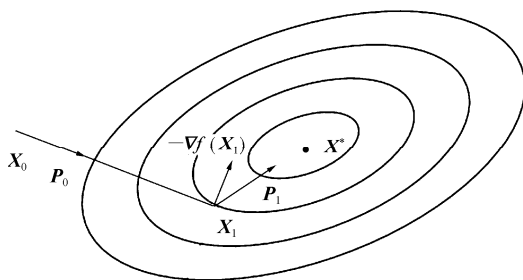


图 3.2.1 共轭梯度法原理

如果按梯度法, 取 $P_1 = -\nabla f(X_1)$, 将使搜索路线出现锯齿现象, 致使算法的收敛速度慢。我们能否取直指极小点 X^* 的方向作为搜索方向 P_1 , 如果能够选定这样的搜索方向, 那么, 对于二维二次函数只需依次进行两次一维搜索, 就可以求得函数的极小点 X^* , 答案是肯定的。下面, 讨论从 X_1 出发直指函数极小点 X^* 的搜索方向应满足什么条件, 如何确定这样的搜索方向。

因为搜索方向 P_1 直指函数的极小点 X^* , 故有

$$X^* = X_1 + \lambda_1 P_1 \quad (3.2.3)$$

式中, λ_1 是最优步长因子, 则有 $f(X_1 + \lambda_1 P_1) = \min f(X_1 + \lambda P_1)$ 。

显然, 当 $X_1 \neq X^*$ 时, $\lambda_1 \neq 0$ 。

对式 (3.2.1) 求梯度, 得

$$\nabla f(X) = AX + B \quad (3.2.4)$$

因为 \mathbf{X}^* 是函数的极小点, 由定理 1.8.1 可知

$$\nabla f(\mathbf{X}^*) = \mathbf{A}\mathbf{X}^* + \mathbf{B} = \mathbf{0}$$

将式 (3.2.3) 代入上式, 得

$$\nabla f(\mathbf{X}_1) + \lambda_1 \mathbf{A}\mathbf{P}_1 = \mathbf{0}$$

上式两边同时左乘 \mathbf{P}_0^T , 并由式 (3.2.2) 和 $\lambda_1 \neq 0$, 得

$$\mathbf{P}_0^T \mathbf{A}\mathbf{P}_1 = 0 \quad (3.2.5)$$

通过上述讨论可知, 过 \mathbf{X}_1 直指函数极小点 \mathbf{X}^* 的搜索方向 \mathbf{P}_1 必须满足式 (3.2.5)。

由于 $\nabla f(\mathbf{X}_1)$ 和 \mathbf{P}_0 正交, 它们可作为二维空间的一组基向量。设

$$\mathbf{P}_1 = -\nabla f(\mathbf{X}_1) + \alpha_0 \mathbf{P}_0 \quad (3.2.6)$$

式中, α_0 为待定常数, 上式两边同时左乘 $\mathbf{P}_0^T \mathbf{A}$, 有

$$\mathbf{P}_0^T \mathbf{A}\mathbf{P}_1 = -\mathbf{P}_0^T \mathbf{A}\nabla f(\mathbf{X}_1) + \alpha_0 \mathbf{P}_0^T \mathbf{A}\mathbf{P}_0 = 0$$

由于 \mathbf{A} 为对称正定矩阵, 有 $\mathbf{P}_0^T \mathbf{A}\mathbf{P}_0 > 0$, 由上式可得

$$\alpha_0 = \frac{\mathbf{P}_0^T \mathbf{A}\nabla f(\mathbf{X}_1)}{\mathbf{P}_0^T \mathbf{A}\mathbf{P}_0}$$

将上式代入式 (3.2.6), 得

$$\mathbf{P}_1 = -\nabla f(\mathbf{X}_1) + \frac{\mathbf{P}_0^T \mathbf{A}\nabla f(\mathbf{X}_1)}{\mathbf{P}_0^T \mathbf{A}\mathbf{P}_0} \mathbf{P}_0 \quad (3.2.7)$$

对于二维二次函数, 从任意初始点 \mathbf{X}_0 出发, 沿任意下降方向 \mathbf{P}_0 作一维搜索得 \mathbf{X}_1 。再从 \mathbf{X}_1 出发, 沿式 (3.2.7) 定义的 \mathbf{P}_1 方向作一维搜索, 所得到的 \mathbf{X}_2 就是二维二次函数 $f(\mathbf{X}) = \frac{1}{2} \mathbf{X}^T \mathbf{A}\mathbf{X} + \mathbf{B}^T \mathbf{X} + c$ 的极小点 \mathbf{X}^* 。

满足式 (3.2.5) 的 \mathbf{P}_0 和 \mathbf{P}_1 向量称为关于对称正定矩阵 \mathbf{A} 的共轭向量。下面, 我们将共轭的概念推广到 n 维空间。

定义 3.2.1 设 \mathbf{A} 为 $n \times n$ 阶对称正定矩阵, 若 n 维空间中非零向量组 $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_{n-1}$ 满足

$$\mathbf{P}_i^T \mathbf{A}\mathbf{P}_j = 0 \quad (i \neq j, i, j = 0, 1, 2, \dots, n-1) \quad (3.2.8)$$

则称向量组 $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_{n-1}$ 关于矩阵 \mathbf{A} 共轭, 或称 $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_{n-1}$ 的方向是关于矩阵 \mathbf{A} 的共轭方向。

当 $\mathbf{A} = \mathbf{I}$ (\mathbf{I} 为 $n \times n$ 阶单位矩阵) 时, 式 (3.2.8) 变为

$$\mathbf{P}_i^T \mathbf{P}_j = 0 \quad (i \neq j, i, j = 0, 1, 2, \dots, n-1)$$

即向量组是相互正交的。由此得知, 共轭向量是正交向量的推广。

设 $\mathbf{A} = \begin{bmatrix} 8 & -4 \\ -4 & 8 \end{bmatrix}$, $\mathbf{P}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\mathbf{P}_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$, 则有

$$\mathbf{P}_0^T \mathbf{A}\mathbf{P}_1 = (1, 0) \begin{bmatrix} 8 & -4 \\ -4 & 8 \end{bmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = 0$$

所以, \mathbf{P}_0 和 \mathbf{P}_1 关于对称正定矩阵 \mathbf{A} 共轭。

同时, 可以验证 $\bar{\mathbf{P}}_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $\bar{\mathbf{P}}_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ 亦是对称正定矩阵 $\mathbf{A} = \begin{bmatrix} 8 & -4 \\ -4 & 8 \end{bmatrix}$ 共轭向量组。由此可

见, 关于对称正定矩阵 A 共轭的向量组不是唯一的。

共轭向量组有如下性质:

性质 1 若向量组 $P_0, P_1, P_2, \dots, P_{n-1}$ 关于 $n \times n$ 阶对称正定矩阵 A 共轭, 则 $P_0, P_1, P_2, \dots, P_{n-1}$ 必定线性无关。

证 用反证法来证明, 假设 $P_0, P_1, P_2, \dots, P_{n-1}$ 线性相关, 则存在一组不全为零的常数 $\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_{n-1}$, 使

$$\alpha_0 P_0 + \alpha_1 P_1 + \alpha_2 P_2 + \dots + \alpha_{n-1} P_{n-1} = 0 \quad (3.2.9)$$

成立。式 (3.2.9) 两边同乘 $P_i^T A$, 由性质 1 的条件, 得

$$\alpha_i P_i^T A P_i = 0 \quad (3.2.10)$$

由于 P_i 不为零向量, 且 A 为对称正定矩阵, 故有

$$P_i^T A P_i > 0 \quad (3.2.11)$$

由式 (3.2.10) 和式 (3.2.11) 得

$$\alpha_i = 0, \quad (i=0, 1, 2, \dots, n-1)$$

上式与假设 $\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_{n-1}$ 不全为零相矛盾, 故性质 1 成立, 证毕。

在给出共轭向量组的性质 2 之前, 首先, 来证明如下引理。

引理 (n 维直交定理) 若向量 $q \in R^n$ 与 n 个线性无关的向量 P_1, P_2, \dots, P_n 都正交, 则 $q=0$ 向量。

证 由于 P_1, P_2, \dots, P_n 线性无关, 它们可作为向量空间 R^n 中的一组基向量, 则

$$q = \alpha_1 P_1 + \alpha_2 P_2 + \dots + \alpha_n P_n$$

由引理的条件和上式, 得

$$\begin{aligned} \|q\|^2 &= q^T q = q^T (\alpha_1 P_1 + \alpha_2 P_2 + \dots + \alpha_n P_n) \\ &= \alpha_1 q^T P_1 + \alpha_2 q^T P_2 + \dots + \alpha_n q^T P_n \\ &= 0 \end{aligned}$$

q 向量的范数为 0, 根据范数的定义, 则有 $q=0$ 向量, 证毕。

(注: 文献[5]对此定理的证明有误, 错误地认为 P_1, P_2, \dots, P_n 相互正交, 上述证明为作者用范数的概念对引理的证明)。

性质 2 如果向量组 $P_0, P_1, P_2, \dots, P_{n-1}$ 关于 $n \times n$ 阶对称正定矩阵 A 共轭, 则从任意初始点 X_0 出发, 相继沿 $P_0, P_1, P_2, \dots, P_{n-1}$ 作 n 次一维搜索, 可得到二次函数 $f(X) = \frac{1}{2} X^T A X + B^T X + c$ 的极小点 X^* , 即迭代点 $X_n = X^*$ 。

证 设 X_k 为第 k 次迭代点, 用一维搜索求最优步长 λ_k , 即

$$f(X_k + \lambda_k P_k) = \min f(X_k + \lambda P_k)$$

引入函数 $\varphi(\lambda) = f(X_k + \lambda P_k)$, 故有

$$\varphi'(\lambda_k) = \nabla f(X_{k+1})^T P_k = 0 \quad (3.2.12)$$

其中, $X_{k+1} = X_k + \lambda_k P_k$ 。

求函数 $f(X) = \frac{1}{2} X^T A X + B^T X + c$ 的梯度

$$\nabla f(X) = A X + B \quad (3.2.13)$$

由式 (3.2.13) 和 $X_{k+1} = X_k + \lambda_k P_k$, 得

$$\nabla f(X_{k+1}) = AX_{k+1} + B = \nabla f(X_k) + \lambda_k AP_k \quad (3.2.14)$$

反复应用式 (3.2.14), 可得

$$\begin{aligned} \nabla f(X_n) &= \nabla f(X_{n-1}) + \lambda_{n-1} AP_{n-1} \\ &= \nabla f(X_{n-2}) + \lambda_{n-2} AP_{n-2} + \lambda_{n-1} AP_{n-1} \\ &\dots\dots \\ &= \nabla f(X_{j+1}) + \sum_{i=j+1}^{n-1} \lambda_i AP_i \end{aligned} \quad (3.2.15)$$

由式 (3.2.15)、式 (3.2.12) 和性质 2 的条件, 有

$$\nabla f(X_n)^T P_j = \nabla f(X_{j+1})^T P_j + \sum_{i=j+1}^{n-1} \lambda_i P_i^T AP_j = 0 \quad (j=0, 1, 2, \dots, n-1)$$

上式说明 $\nabla f(X_n)$ 与 n 个线性无关的向量组正交, 由上式和引理可知

$$\nabla f(X_n) = 0$$

故有 $X_n = X^*$, 证毕。

由性质 2 可知, 如果能找到关于 $n \times n$ 阶对称正定矩阵 A 的一组共轭向量, 则经过 n 次一维搜索, 就能求得二次函数 $f(X) = \frac{1}{2} X^T AX + B^T X + c$ 的极小点 X^* 。现在的问题是如何寻找一组关于 A 矩阵共轭的向量, 下面的定理将告诉我们如何求得一组共轭向量。

定理 3.2.1 考虑二次函数最优化问题 $\min f(X) = \frac{1}{2} X^T AX + B^T X + c$, 其中, A 为 $n \times n$

阶对称正定矩阵, $X \in R^n$ 。设 X_0 为任意初始点, 令 $P_0 = -\nabla f(X_0)$, 设 λ_k ($k=0, 1, 2, \dots, n-1$) 是

$\min f(X_k + \lambda P_k)$ 的最优解。令 $X_{k+1} = X_k + \lambda_k P_k$ 和 $P_{k+1} = -\nabla f(X_{k+1}) + \alpha_k P_k$, 其中, $\alpha_k = \frac{\|\nabla f(X_{k+1})\|^2}{\|\nabla f(X_k)\|^2}$,

则

(1) $P_0, P_1, P_2, \dots, P_{n-1}$ 是关于矩阵 A 共轭的向量组。

(2) $P_0, P_1, P_2, \dots, P_{n-1}$ 是函数分别在 $X_0, X_1, X_2, \dots, X_{n-1}$ 处的下降方向。

$$(3) \quad \alpha_k = \frac{\|\nabla f(X_{k+1})\|^2}{\|\nabla f(X_k)\|^2} = \frac{P_k^T A \nabla f(X_{k+1})}{P_k^T A P_k} \quad (k=0, 1, 2, \dots, n-1)。$$

(4) $\nabla f(X_i)^T \nabla f(X_j) = 0 \quad (i \neq j, 0 \leq i, j \leq n-1)。$

证 引入辅助函数 $\varphi(\lambda) = f(X_k + \lambda P_k)$, 有

$$\varphi'(\lambda_k) = \nabla f(X_{k+1})^T P_k = 0 \quad (3.2.16)$$

由于 $f(X) = \frac{1}{2} X^T AX + B^T X + c$, 有

$$\nabla f(X_{k+1}) = \nabla f(X_k) + \lambda_k AP_k \quad (3.2.17)$$

由式 (3.2.16)、式 (3.2.17) 和 $P_k = -\nabla f(X_k) + \alpha_{k-1} P_{k-1}$, 得

$$\nabla f(X_{k+1})^T P_k = [\nabla f(X_k) + \lambda_k AP_k]^T P_k$$

$$\begin{aligned}
&= \nabla f(X_k)^T P_k + \lambda_k P_k^T A P_k \\
&= \nabla f(X_k)^T [-\nabla f(X_k) + \alpha_{k-1} P_{k-1}] + \lambda_k P_k^T A P_k \\
&= -\nabla f(X_k)^T \nabla f(X_k) + \lambda_k P_k^T A P_k = 0
\end{aligned}$$

暂时假设 $P_k \neq 0$ 向量, 由上式可得

$$\lambda_k = \frac{\nabla f(X_k)^T \nabla f(X_k)}{P_k^T A P_k} \quad (3.2.18)$$

首先, 用归纳法证明:

$$\textcircled{1} \quad P_i^T A P_j = 0 \quad (i \neq j, 0 \leq i, j \leq n-1)$$

$$\textcircled{2} \quad \nabla f(X_i)^T \nabla f(X_j) = 0 \quad (i \neq j, 0 \leq i, j \leq n-1)$$

$$\textcircled{3} \quad \alpha_i = \frac{P_i^T A \nabla f(X_{i+1})}{P_i^T A P_i} = \frac{\|\nabla f(X_{i+1})\|^2}{\|\nabla f(X_i)\|^2} \quad (i=0, 1, 2, \dots, n-1)$$

第一步验证当 $i=0, j=1$ 时, ①、②和③成立。

$$A P_k = \frac{1}{\lambda_k} A(X_{k+1} - X_k) = \frac{1}{\lambda_k} [\nabla f(X_{k+1}) - \nabla f(X_k)] \quad (3.2.19)$$

由式 (3.2.16)、式 (3.2.19) 和 $P_0 = -\nabla f(X_0)$, 得

$$\begin{aligned}
\alpha_0 &= \frac{P_0^T A \nabla f(X_1)}{P_0^T A P_0} \\
&= \frac{\nabla f(X_1)^T [\nabla f(X_1) - \nabla f(X_0)]}{P_0^T [\nabla f(X_1) - \nabla f(X_0)]} = \frac{\|\nabla f(X_1)\|^2}{\|\nabla f(X_0)\|^2}
\end{aligned} \quad (3.2.20)$$

$$\nabla f(X_0)^T \nabla f(X_1) = -\nabla f(X_1)^T P_0 = 0 \quad (3.2.21)$$

$$\begin{aligned}
P_0^T A P_1 &= P_0^T A [-\nabla f(X_1) + \alpha_0 P_0] \\
&= -P_0^T A \nabla f(X_1) + \alpha_0 P_0^T A P_0 = 0
\end{aligned} \quad (3.2.22)$$

以上三式验证了当 $i=0, j=1$ 时, ①、②和③成立。

第二步假设①、②和③对于某个 $k(k < n-1)$ 成立, 即

$$P_i^T A P_j = 0 \quad (i \neq j, 0 \leq i, j \leq k) \quad (3.2.23)$$

$$\nabla f(X_i)^T \nabla f(X_j) = 0 \quad (i \neq j, 0 \leq i, j \leq k) \quad (3.2.24)$$

$$\alpha_i = \frac{P_i^T A \nabla f(X_{i+1})}{P_i^T A P_i} = \frac{\|\nabla f(X_{i+1})\|^2}{\|\nabla f(X_i)\|^2} \quad (i=0, 1, 2, \dots, k-1) \quad (3.2.25)$$

第三步证明对于 $k+1$, 上面三式成立。

对于 $i \in (1, 2, \dots, k-1)$

$$\begin{aligned}
\nabla f(X_{k+1})^T \nabla f(X_i) &= [\nabla f(X_k) + \lambda_k A P_k]^T \nabla f(X_i) \\
&= \lambda_k P_k^T A \nabla f(X_i)
\end{aligned}$$

$$= \lambda_k \mathbf{P}_k^T \mathbf{A} (-\mathbf{P}_i + \alpha_{i-1} \mathbf{P}_{i-1}) = 0 \quad (3.2.26)$$

$$\begin{aligned} \nabla f(\mathbf{X}_{k+1})^T \nabla f(\mathbf{X}_k) &= [\nabla f(\mathbf{X}_k) + \lambda_k \mathbf{A} \mathbf{P}_k]^T \nabla f(\mathbf{X}_k) \\ &= \|\nabla f(\mathbf{X}_k)\|^2 + \lambda_k \mathbf{P}_k^T \mathbf{A} (-\mathbf{P}_k + \alpha_{k-1} \mathbf{P}_{k-1}) \\ &= \|\nabla f(\mathbf{X}_k)\|^2 - \lambda_k \mathbf{P}_k^T \mathbf{A} \mathbf{P}_k = 0 \end{aligned} \quad (3.2.27)$$

$$\begin{aligned} \nabla f(\mathbf{X}_{k+1})^T \nabla f(\mathbf{X}_0) &= [\nabla f(\mathbf{X}_k) + \lambda_k \mathbf{A} \mathbf{P}_k]^T \nabla f(\mathbf{X}_0) \\ &= \nabla f(\mathbf{X}_k)^T \nabla f(\mathbf{X}_0) - \lambda_k \mathbf{P}_k^T \mathbf{A} \mathbf{P}_0 = 0 \end{aligned} \quad (3.2.28)$$

上面三式说明②成立，即定理 3.2.1 结论 (4) 成立。

由式 (3.2.17) 和 $\mathbf{P}_k = -\nabla f(\mathbf{X}_k) + \alpha_{k-1} \mathbf{P}_{k-1}$ ，得

$$\begin{aligned} \alpha_k &= \frac{\nabla f(\mathbf{X}_{k+1})^T \mathbf{A} \mathbf{P}_k}{\mathbf{P}_k^T \mathbf{A} \mathbf{P}_k} = \frac{\nabla f(\mathbf{X}_{k+1})^T [\nabla f(\mathbf{X}_{k+1}) - \nabla f(\mathbf{X}_k)]}{\mathbf{P}_k^T [\nabla f(\mathbf{X}_{k+1}) - \nabla f(\mathbf{X}_k)]} \\ &= \frac{\|\nabla f(\mathbf{X}_{k+1})\|^2}{\|\nabla f(\mathbf{X}_k)\|^2} \end{aligned} \quad (3.2.29)$$

故③成立，即定理 3.2.1 结论 (3) 成立。

对于 $i \in (0, 1, 2, \dots, k-1)$

$$\begin{aligned} \mathbf{P}_{k+1}^T \mathbf{A} \mathbf{P}_i &= [-\nabla f(\mathbf{X}_{k+1}) + \alpha_k \mathbf{P}_k]^T \mathbf{A} \mathbf{P}_i \\ &= -\nabla f(\mathbf{X}_{k+1})^T \mathbf{A} \mathbf{P}_i \\ &= -\frac{1}{\lambda_i} \nabla f(\mathbf{X}_{k+1})^T [\nabla f(\mathbf{X}_{i+1}) - \nabla f(\mathbf{X}_i)] = 0 \end{aligned} \quad (3.2.30)$$

$$\mathbf{P}_{k+1}^T \mathbf{A} \mathbf{P}_k = [-\nabla f(\mathbf{X}_{k+1}) + \alpha_k \mathbf{P}_k]^T \mathbf{A} \mathbf{P}_k = 0 \quad (3.2.31)$$

由以上两式可知①成立。

要证明定理 3.2.1 结论 (1) 成立，只要证 $\mathbf{P}_i \neq 0$ 向量 ($i=0, 1, 2, \dots, n-1$)。

$$\|\mathbf{P}_0\|^2 = \|\nabla f(\mathbf{X}_0)\|^2 > 0 \quad (3.2.32)$$

$$\begin{aligned} \|\mathbf{P}_i\|^2 &= [-\nabla f(\mathbf{X}_i) + \alpha_{i-1} \mathbf{P}_{i-1}]^T [-\nabla f(\mathbf{X}_i) + \alpha_{i-1} \mathbf{P}_{i-1}] \\ &= \|\nabla f(\mathbf{X}_i)\|^2 + \alpha_{i-1}^2 \|\mathbf{P}_{i-1}\|^2 > 0 \quad (i=0, 1, 2, \dots, n-1) \end{aligned} \quad (3.2.33)$$

故定理 3.2.1 结论 (1) 成立。

显然， $\mathbf{P}_0 = -\nabla f(\mathbf{X}_0)$ 为函数在 \mathbf{X}_0 处的下降方向。

$$\begin{aligned} \nabla f(\mathbf{X}_i)^T \mathbf{P}_i &= \nabla f(\mathbf{X}_i)^T [-\nabla f(\mathbf{X}_i) + \alpha_{i-1} \mathbf{P}_{i-1}] \\ &= -\|\nabla f(\mathbf{X}_i)\|^2 < 0 \quad (i=0, 1, 2, \dots, n-1) \end{aligned} \quad (3.2.34)$$

上式说明了 \mathbf{P}_i 为函数在 \mathbf{X}_i 处的下降方向。故定理 3.2.1 结论 (2) 成立，证毕。

推论 对于非二次函数，定理 3.2.1 中的 $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_{n-1}$ 仍为函数的下降方向。

由于最优步长 λ_k 的求解产生很小的误差和计算时的舍入误差，这些误差的积累有可能致

使经过 n 次一维搜索不能求得二次函数的极小点 \mathbf{X}^* 。同时, 为了使共轭梯度法能用于非二次函数, 因此, n 次一维搜索后需要继续迭代。由于共轭向量组是线性无关的, 而 \mathbf{R}^n 中线性无关的向量最多只有 n 。因此, 在实际应用中, 再令 \mathbf{X}_n 为初始点, 重新进行迭代。

共轭梯度法的主要步骤如下:

(1) 给定初始点 \mathbf{X}_0 和收敛精度 ε 。

(2) 置 $k=0$, $\mathbf{P}_0 = -\nabla f(\mathbf{X}_0)$ 。

(3) 求最优步长因子 λ_0 , 即使 $f(\mathbf{X}_0 + \lambda_0 \mathbf{P}_0) = \min f(\mathbf{X}_0 + \lambda \mathbf{P}_0)$, 置 $\mathbf{X}_1 = \mathbf{X}_0 + \lambda_0 \mathbf{P}_0$ 。

(4) 设 \mathbf{X}_k 为第 k 次迭代点, 令 $\mathbf{P}_k = -\nabla f(\mathbf{X}_k) + \alpha_{k-1} \mathbf{P}_{k-1}$, 其中 $\alpha_{k-1} = \frac{\|\nabla f(\mathbf{X}_k)\|^2}{\|\nabla f(\mathbf{X}_{k-1})\|^2}$ 。

(5) 求最优步长因子 λ_k , 即使 $f(\mathbf{X}_k + \lambda_k \mathbf{P}_k) = \min f(\mathbf{X}_k + \lambda \mathbf{P}_k)$, 置 $\mathbf{X}_{k+1} = \mathbf{X}_k + \lambda_k \mathbf{P}_k$, $k = k+1$ 。

(6) 若 $\|\nabla f(\mathbf{X}_{k+1})\| < \varepsilon$, 置 $\mathbf{X}^* = \mathbf{X}_{k+1}$, 并打印最优解 \mathbf{X}^* 和 $f(\mathbf{X}^*)$, 停止程序执行。

(7) 若 $k=n$, 置 $\mathbf{X}_0 = \mathbf{X}_n$, 转到步骤 (2); 否则, 转到步骤 (4)。

共轭梯度法程序框图如图 3.2.2 所示。

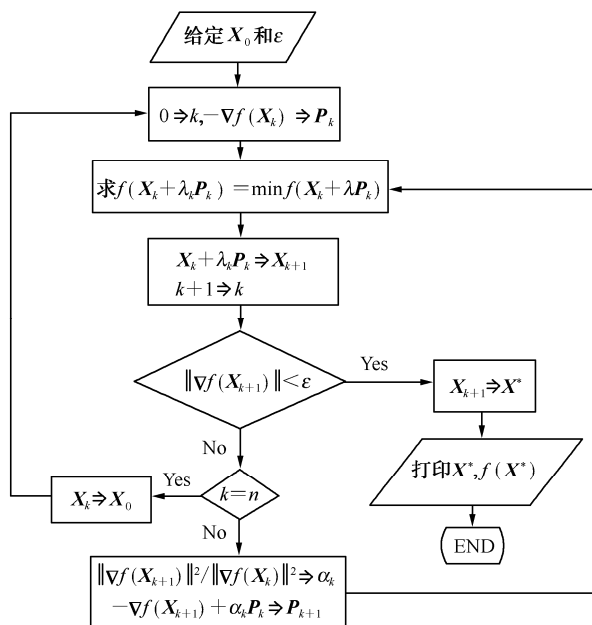


图 3.2.2 共轭梯度法程序框图

例 3.2.1 用共轭梯度法求函数 $f(\mathbf{X}) = x_1^2 + 25x_2^2$ 的极小点, 取初始点 $\mathbf{X}_0 = (70, 9)^T$ 。

解

$$\nabla f(\mathbf{X}) = (2x_1, 50x_2)^T$$

$$\mathbf{P}_0 = -\nabla f(\mathbf{X}_0) = (-140, -450)^T$$

$$\mathbf{X} = \mathbf{X}_0 + \lambda \mathbf{P}_0 = \begin{pmatrix} 70 - 140\lambda \\ 9 - 450\lambda \end{pmatrix}$$

令 $\varphi(\lambda)=f(\mathbf{X}_0+\lambda\mathbf{P}_0)=(70-140\lambda)^2+25(9-450\lambda)^2$
求最优步长因子 λ_0 , 即使 $\varphi(\lambda_0)=\min \varphi(\lambda)$ 。令 $\varphi'(\lambda_0)=0$, 得 $\lambda_0=0.0218512032$

$$\mathbf{X}_1=\mathbf{X}_0+\lambda_0\mathbf{P}_0=(66.94083155, -0.83304144)^T$$

$$\nabla f(\mathbf{X}_1)=(133.8816631, -41.652072)^T$$

$$\alpha_0=\frac{\|\nabla f(\mathbf{X}_1)\|^2}{\|\nabla f(\mathbf{X}_0)\|^2}=0.0885150599$$

$$\mathbf{P}_1=-\nabla f(\mathbf{X}_1)+\alpha_0\mathbf{P}_0=(-146.2737714, 1.820295045)^T$$

$$\mathbf{X}=\mathbf{X}_1+\lambda\mathbf{P}_1=\begin{pmatrix} 66.94083155-146.2737714\lambda \\ -0.83304144+1.820295045\lambda \end{pmatrix}$$

令 $\varphi(\lambda)=f(\mathbf{X}_1+\lambda\mathbf{P}_1)=(66.94083155-146.2737714\lambda)^2+25(-0.83304144+1.820295045\lambda)^2$

求最优步长因子 λ_1 , 即使 $\varphi(\lambda_1)=\min \varphi(\lambda)$ 。令 $\varphi'(\lambda_1)=0$, 得 $\lambda_1=0.4576407035$

$$\mathbf{X}_2=\mathbf{X}_1+\lambda_1\mathbf{P}_1=(-9.7\times 10^{-8}, 3.35\times 10^{-7})^T$$

本例题的精确解 $\mathbf{X}^*=(0, 0)^T$, 数例验证了定理 3.2.1 和共轭向量组性质 2 的正确性。

作者用抛物线插值法作为一维搜索方法, 编写了共轭梯度法程序, 用程序计算的有关数据如表 3.2.1 所示。

表 3.2.1 共轭梯度法计算的有关数据

| 迭代次数 | x_1^k | x_2^k | $\frac{\partial f(\mathbf{X}_k)}{\partial x_1}$ | $\frac{\partial f(\mathbf{X}_k)}{\partial x_2}$ | λ_k |
|------|------------------------|-----------------------|---|---|-------------|
| 0 | 70 | 9 | 140 | 450 | 0.021851 |
| 1 | 66.94083 | 0.833042 | 133.8817 | -41.65211 | 0.45764 |
| 2 | 7.6829×10^{-6} | 1.556×10^{-5} | 1.526×10^{-5} | 7.778×10^{-4} | 0.02001 |

3.3 牛顿法

如果目标函数 $f(\mathbf{X})$ 具有连续二阶偏导数, 在 \mathbf{X}_k 处将函数二阶 Taylor 展开, 得

$$f(\mathbf{X})\approx q(\mathbf{X})=f(\mathbf{X}_k)+\nabla f(\mathbf{X}_k)^T(\mathbf{X}-\mathbf{X}_k)+\frac{1}{2}(\mathbf{X}-\mathbf{X}_k)^T\mathbf{H}(\mathbf{X}_k)(\mathbf{X}-\mathbf{X}_k) \quad (3.3.1)$$

在上式中, 如果 Hessian 矩阵 $\mathbf{H}(\mathbf{X}_k)$ 是非奇异矩阵, 可求得二次函数 $q(\mathbf{X})$ 的极小点。对插值函数 $q(\mathbf{X})$ 求梯度, 得

$$\nabla q(\mathbf{X})=\nabla f(\mathbf{X}_k)+\mathbf{H}(\mathbf{X}_k)(\mathbf{X}-\mathbf{X}_k)=0$$

解上式, 并将 $q(\mathbf{X})$ 的极小点作为迭代点 \mathbf{X}_{k+1} , 可得

$$\mathbf{X}_{k+1}=\mathbf{X}_k-\mathbf{H}(\mathbf{X}_k)^{-1}\nabla f(\mathbf{X}_k) \quad (3.3.2)$$

式 (3.3.2) 就是牛顿迭代公式。可视为搜索方向 $\mathbf{P}_k=-\mathbf{H}(\mathbf{X}_k)^{-1}\nabla f(\mathbf{X}_k)$ 和最优步长因子 $\lambda_k=1$ 。

如果目标函数 $f(\mathbf{X})$ 是二次函数, 则有 $f(\mathbf{X})=q(\mathbf{X})$, 从任意的 \mathbf{X}_0 出发, 用式 (3.3.2) 计算, 只需一步就能求得 $f(\mathbf{X})$ 的极小点。

例 3.3.1 用牛顿法求函数 $f(\mathbf{X})=x_1^2+25x_2^2$ 的极小点, 取初始点 $\mathbf{X}_0=(70, 9)^T$ 。

解 将函数用矩阵向量表示

$$f(\mathbf{X})=x_1^2+25x_2^2=\frac{1}{2}(x_1, x_2)\begin{bmatrix} 2 & 0 \\ 0 & 50 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\nabla f(\mathbf{X}_0)=\begin{pmatrix} 140 \\ 450 \end{pmatrix}$$

$$\mathbf{H}(\mathbf{X}_0)=\begin{bmatrix} 2 & 0 \\ 0 & 50 \end{bmatrix}, \quad \mathbf{H}(\mathbf{X}_0)^{-1}=\begin{bmatrix} 1/2 & 0 \\ 0 & 1/50 \end{bmatrix}$$

$$\mathbf{X}_1=\mathbf{X}_0-\mathbf{H}(\mathbf{X}_0)^{-1}\nabla f(\mathbf{X}_0)=\begin{pmatrix} 70 \\ 9 \end{pmatrix}-\begin{pmatrix} 1/2 & 0 \\ 0 & 1/50 \end{pmatrix}\begin{pmatrix} 140 \\ 450 \end{pmatrix}=\begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

即 $\mathbf{X}_1=\mathbf{X}^*$ 。

大量的实例计算已经证明牛顿法不适合于非二次函数, 这是因为当初始点 \mathbf{X}_0 远离极小点 \mathbf{X}^* 时, 牛顿法的步长很大, 有可能使迭代序列发散。因此, 通常对牛顿法作如下改进。

1. 如果在某迭代点 \mathbf{X}_k 处的 Hessian 矩阵 $\mathbf{H}(\mathbf{X}_k)$ 为非奇异矩阵, 令 $\mathbf{P}_k=-\mathbf{H}(\mathbf{X}_k)^{-1}\nabla f(\mathbf{X}_k)$ 。如果 \mathbf{P}_k 为函数在 \mathbf{X}_k 处的下降方向, 沿 \mathbf{P}_k 作一维搜索求得最优步长因子 λ_k 。

$$\text{令} \quad \mathbf{X}_{k+1}=\mathbf{X}_k+\lambda_k\mathbf{P}_k \quad (3.3.3)$$

我们称 $\mathbf{P}_k=-\mathbf{H}(\mathbf{X}_k)^{-1}\nabla f(\mathbf{X}_k)$ 为牛顿方向。

2. 如果 $\mathbf{P}_k=-\mathbf{H}(\mathbf{X}_k)^{-1}\nabla f(\mathbf{X}_k)$ 不是函数的下降方向, 按梯度法求解。我们用下例来说明牛顿方向不是函数的下降方向。

例 3.3.2 用牛顿法求函数 $f(\mathbf{X})=x_1^4+x_1x_2+(1+x_2)^2$ 的极小点, 取初始点 $\mathbf{X}_0=(0, 0)^T$ 。

解 求函数 $f(\mathbf{X})=x_1^4+x_1x_2+(1+x_2)^2$ 在 \mathbf{X}_0 处的梯度和 Hessian 矩阵, 得

$$\nabla f(\mathbf{X}_0)=\begin{pmatrix} 4x_1^3+x_2 \\ x_1+2x_2+2 \end{pmatrix}=\begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

$$\mathbf{H}(\mathbf{X}_0)=\begin{bmatrix} 12x_1^2 & 1 \\ 1 & 2 \end{bmatrix}=\begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix}$$

求 $\mathbf{H}(\mathbf{X}_0)$ 的逆矩阵, 得

$$\mathbf{H}(\mathbf{X}_0)^{-1}=\begin{bmatrix} -2 & 1 \\ 1 & 0 \end{bmatrix}$$

所以

$$\mathbf{P}_0=-\mathbf{H}(\mathbf{X}_0)^{-1}\nabla f(\mathbf{X}_0)=\begin{bmatrix} -2 & 1 \\ 1 & 0 \end{bmatrix}\begin{pmatrix} 0 \\ 2 \end{pmatrix}=\begin{pmatrix} -2 \\ 0 \end{pmatrix}$$

令

$$\varphi(\lambda)=f(\mathbf{X}_0+\lambda\mathbf{P}_0)=16\lambda^4+1$$

求函数 $\varphi(\lambda)$ 的驻点

$$\varphi'(\lambda)=64\lambda^3=0$$

由上式得

$$\lambda_0=0$$

故有

$$\mathbf{X}_1=\mathbf{X}_0+\lambda_0\mathbf{P}_0=\mathbf{X}_0$$

例题说明函数的 Hessian 矩阵 $\mathbf{H}(\mathbf{X}_k)$ 即使为非奇异矩阵, 也求不到新的迭代点。在这种情况下, 按负梯度方向求新的迭代 \mathbf{X}_{k+1} , 即取 $\mathbf{P}_k=-\nabla f(\mathbf{X}_k)$, 再用一维搜索方法求最优步长因子。

3. 如果 Hessian 矩阵 $\mathbf{H}(\mathbf{X}_k)$ 为奇异矩阵, 我们也按梯度法求解。

牛顿法的收敛速度是最快的。可以证明当函数满足一定的条件, 牛顿法为二阶收敛, 这是牛顿法的优点。但是牛顿法要求 Hessian 矩阵 $\mathbf{H}(\mathbf{X}_k)$ 为正定矩阵。其次, 求 Hessian 矩阵的逆矩阵 $\mathbf{H}(\mathbf{X}_k)^{-1}$ 计算量与优化问题的维数 n 的三次幂 (n^3) 成比例, 使得每一次迭代要花较长的时间, 这又抵消了它收敛速度快的优点。

由于一般函数在极小点附近呈二次函数, 所以, 在极小点 \mathbf{X}^* 附近用牛顿法能充分利用其收敛速度快的优点。同时, 当迭代点远离极小点 \mathbf{X}^* 时, 用梯度法是比较有利的。故一般是梯度法与牛顿法混合使用, 即首先用梯度法计算, 使迭代点靠近极小点 \mathbf{X}^* ; 再用牛顿法计算。这样能充分地利用这两种方法的优点, 又能避免这两种方法的缺点。

牛顿法是最著名的基本的最优化方法之一, 由牛顿法的思想发展起来了一些更有效的其他算法。下面, 叙述由牛顿法的思想发展起来的变尺度法 (或称为拟牛顿法)。

3.4 变尺度法

变尺度法既避免了牛顿法计算 Hessian 矩阵 $\mathbf{H}(\mathbf{X}_k)$ 逆矩阵的过程, 又比共轭梯度法有更好的收敛速度, 它对高维最优化问题的求解具有明显的优越性。变尺度法曾经获得了很高的声誉, 被认为是求解最优化问题的最有效的方法之一。

3.4.1 变尺度法的基本思想

所谓尺度就是度量, 在向量空间 \mathbf{R}^n 中, 最常用的度量是用如下的范数:

$$\|\mathbf{X}\|=(\mathbf{X}^T\mathbf{X})^{1/2} \quad (\mathbf{X}\in\mathbf{R}^n) \quad (3.4.1)$$

本章 3.1 节讨论的梯度法, 就是在度量 (3.4.1) 意义下, 确定了函数 $f(\mathbf{X})$ 在 \mathbf{X}_k 处的最速下降方向为 $\mathbf{P}_k=-\nabla f(\mathbf{X}_k)$ 。从局部意义上讲, 它的确是函数 $f(\mathbf{X})$ 在 \mathbf{X}_k 处最速下降方向。然而, 从整体来看它不仅不是最优的, 反而使搜索过程绕了许多弯路, 在极小点附近尤其如此, 这是导致梯度法收敛速度慢的根本原因。

那么, 是否可以选择另一种度量, 从整体意义上确定更好的搜索方向呢? 为了更清楚地说明这个问题, 我们用牛顿法求解如下二次函数的优化问题:

$$\min f(\mathbf{X})$$

其中, $f(\mathbf{X})=\frac{1}{2}\mathbf{X}^T\mathbf{A}\mathbf{X}+\mathbf{B}^T\mathbf{X}+c$, \mathbf{A} 为 $n\times n$ 阶对称正定矩阵, \mathbf{B} 为 n 维向量, c 为常数。

由上式可得

$$X^* = X_0 - H(X_0)^{-1} \nabla f(X_0) = -A^{-1}B$$

其中, X_0 为任意初始点。

若取搜索方向为

$$P_0 = -H(X_0)^{-1} \nabla f(X_0)$$

则进行一次一维搜索就能得到二次函数的极小点。

我们利用 $n \times n$ 阶对称正定矩阵 A , 引进一种新的度量

$$\|X\| = (X^T A X)^{1/2} \quad (X \in R^n) \quad (3.4.2)$$

在第1章1.3节中, 我们已经证明了 $\|X\| = (X^T A X)^{1/2}$ 是一种范数, 即满足范数的3个条件:

(1) $\|X\| \geq 0$, 当且仅当 $X=0$ 向量时, $\|X\|=0$

(2) $\|\alpha X\| = |\alpha| \|X\|$, 对任意常数 $\alpha \in K$

(3) $\|X+Y\| \leq \|X\| + \|Y\|$

且有

$$X^T A Y \leq (X^T A X)^{1/2} (Y^T A Y)^{1/2} = \|X\| \|Y\| \quad (3.4.3)$$

如果没有特别说明, 下面本节中使用的向量的范数是指按式(3.4.2)定义的范数。

在式(3.4.2)尺度的意义下, 我们来证明对于二次函数 $f(X) = \frac{1}{2} X^T A X + B^T X + c$, 搜索方

向 $P_0 = -A^{-1} \nabla f(X_0)$ 就是函数的最速下降方向。

$$\begin{aligned} f(X_0 + P_0) - f(X_0) &= \nabla f(X_0)^T P_0 + \frac{1}{2} P_0^T A P_0 \\ &= \nabla f(X_0)^T A^{-1} A P_0 + \frac{1}{2} P_0^T A P_0 \\ &\geq -\|A^{-1} \nabla f(X_0)\| \|P_0\| + \frac{1}{2} \|P_0\|^2 \end{aligned}$$

令 $\alpha = \|P_0\|$, 上式右边是 α 的函数。令

$$\varphi(\alpha) = -\|A^{-1} \nabla f(X_0)\| \alpha + \frac{1}{2} \alpha^2$$

由上式, 得

$$\begin{aligned} \varphi'(\alpha^*) &= -\|A^{-1} \nabla f(X_0)\| + \alpha^* = 0 \\ \alpha^* &= \|A^{-1} \nabla f(X_0)\| \end{aligned}$$

故 $P_0 = -A^{-1} \nabla f(X_0)$ 是在由式(3.4.2)定义度量的意义下的最速下降方向。

我们考虑改进的牛顿法的迭代公式

$$X_{k+1} = X_k - \lambda_k H(X_k)^{-1} \nabla f(X_k) \quad (3.4.4)$$

为了避免求 $H(X_k)$ 的逆矩阵, 我们用某种近似矩阵 $A_k = A(X_k)$ 来近似代替 $H(X_k)^{-1}$, 则上式变为

$$X_{k+1} = X_k - \lambda_k A_k \nabla f(X_k) \quad (3.4.5)$$

当 $A_k = I$ (I 为单位矩阵) 时, 式(3.4.5)为梯度法的迭代公式。当 $A_k = H(X_k)^{-1}$ 时, 式(3.4.5)为改进的牛顿法的迭代公式。

在式(3.4.5)中, 由于 A_k 在迭代过程中是变化的, 我们度量最速下降的尺度是变化的,

故以式 (3.4.5) 作为迭代公式的算法称为变尺度法; 又由于式 (3.4.5) 在形式上与牛顿法相同, 故又称为拟牛顿法。

现在, 先来讨论构造 A_k 的几个原则。我们希望能使算法具有下降性质、计算方便, 并能保证算法有好的收敛性, 因此, 构造 A_k 的几个原则为:

(1) 为了保证搜索方向 $P_k = -A_k \nabla f(X_k)$ 为函数在 X_k 处的下降方向, 要求 A_k 为对称正定矩阵。如果 A_k 为对称正定矩阵, 有

$$\nabla f(X_k)^T P_k = -\nabla f(X_k)^T A_k \nabla f(X_k) < 0 \quad (3.4.6)$$

(2) 为了保证变尺度法对于二次函数具有有限步收敛的性质, 要求搜索方向组为共轭方向组, 即

$$P_{k+1}^T A P_j = 0 \quad (0 \leq j \leq k, k=1, 2, \dots) \quad (3.4.7)$$

$$\nabla f(X_{k+1})^T P_j = 0 \quad (0 \leq j \leq k, k=1, 2, \dots) \quad (3.4.8)$$

不难由定理 3.2.1 结论 (4) 证明式 (3.4.8)。

(3) 为了计算方便, 希望尺度矩阵 A_k 具有如下递推形式:

$$A_{k+1} = A_k + E_k \quad (3.4.9)$$

其中, E_k 为修正矩阵。修正矩阵 E_k 的确定, 要求只依赖于当前已经求得的迭代点 X_k, X_{k+1} 和它们相应的函数的梯度。

3.4.2 一族变尺度法的形成

我们假设式 (3.4.7) 和式 (3.4.8) 对于 $j \leq k-1$ 成立, 现在来推导 A_{k+1} 。

由式 (3.4.5) 知, 搜索方向为

$$P_k = -A_k \nabla f(X_k) \quad (3.4.10)$$

将式 (3.4.10) 代入式 (3.4.7), 得

$$P_{k+1}^T A P_j = -\nabla f(X_{k+1})^T A_{k+1} A P_j = 0 \quad (0 \leq j \leq k) \quad (3.4.11)$$

由于 $\nabla f(X_{k+1}) \neq 0$, 由式 (3.4.11) 容易看出, 要满足式 (3.4.7) 和式 (3.4.8), 只需要搜索方向 P_j 满足

$$A_{k+1} A P_j = \rho P_j \quad (0 \leq j \leq k) \quad (3.4.12)$$

其中, ρ 为任意实数。

由矩阵 A_{k+1} 的构造, 将式 (3.4.9) 代入式 (3.4.12), 并假设式 (3.4.12) 对于 $0 \leq j \leq k-1$ 成立, 则得

$$E_k^T A P_j = \rho P_j - A_k A P_j = 0 \quad (0 \leq j \leq k-1) \quad (3.4.13)$$

对于 $j=k$, 将式 (3.4.9) 代入式 (3.4.12) 得

$$E_k^T A P_k = \rho P_k - A_k A P_k \quad (3.4.14)$$

令

$$S_j = X_{j+1} - X_j = \lambda_j P_j \quad (\lambda_j \neq 0) \quad (3.4.15)$$

$$Y_j = \nabla f(X_{j+1}) - \nabla f(X_j) = A S_j = \lambda_j A P_j \quad (3.4.16)$$

由上面两式, 式 (3.4.13) 和式 (3.4.14) 可变为

$$E_k Y_j = 0 \quad (0 \leq j \leq k-1) \quad (3.4.17)$$

$$E_k Y_k = \rho S_k - A_k Y_k \quad (3.4.18)$$

式(3.4.17)和式(3.4.18)就是修正矩阵 \mathbf{E}_k 应满足的方程式。

用式(3.4.17)和式(3.4.18)确定 \mathbf{E}_k , 我们取修正矩阵

$$\mathbf{E}_k = \mathbf{S}_k \mathbf{U}_k^T + \mathbf{A}_k \mathbf{Y}_k \mathbf{V}_k^T \quad (3.4.19)$$

其中, $\mathbf{U}_k, \mathbf{V}_k \in \mathbf{R}^n$, 为待定向量。

为了使式(3.4.19)满足式(3.4.17)和式(3.4.18), 我们分别取

$$\mathbf{U}_k^T \mathbf{Y}_j = \begin{cases} 0 & (0 \leq j \leq k-1) \\ \rho & (j=k) \end{cases} \quad (3.4.20)$$

$$\mathbf{V}_k^T \mathbf{Y}_j = \begin{cases} 0 & (0 \leq j \leq k-1) \\ -1 & (j=k) \end{cases} \quad (3.4.21)$$

由于已经假定式(3.4.7)对于 $j \leq k-1$ 成立, 有

$$\mathbf{P}_k^T \mathbf{A} \mathbf{P}_j = 0 \quad (0 \leq j \leq k-1) \quad (3.4.22)$$

由式(3.4.16), 可得

$$\mathbf{A} \mathbf{P}_j = \frac{\mathbf{Y}_j}{\lambda_j} \quad (0 \leq j \leq k-1) \quad (3.4.23)$$

将式(3.4.23)和式(3.4.15)代入式(3.4.22), 得

$$\mathbf{S}_k^T \mathbf{Y}_j = 0 \quad (0 \leq j \leq k-1) \quad (3.4.24)$$

由式(3.4.8)和式(3.4.16), 可得

$$\mathbf{Y}_k^T \mathbf{P}_j = 0 \quad (0 \leq j \leq k-1) \quad (3.4.25)$$

由上式和式(3.4.12), 得

$$\mathbf{Y}_k^T \mathbf{A}_k \mathbf{A} \mathbf{P}_j = 0 \quad (0 \leq j \leq k-1) \quad (3.4.26)$$

由上式和式(3.4.23), 得

$$\mathbf{Y}_k^T \mathbf{A}_k \mathbf{Y}_j = 0 \quad (0 \leq j \leq k-1) \quad (3.4.27)$$

比较式(3.4.20)、式(3.4.21)、式(3.4.24)和式(3.4.27), 可看出向量 \mathbf{U}_k 和 \mathbf{V}_k 可选为

$$\mathbf{U}_k = a_{11}^k \mathbf{S}_k + a_{12}^k \mathbf{A}_k^T \mathbf{Y}_k \quad (3.4.28)$$

$$\mathbf{V}_k = a_{21}^k \mathbf{S}_k + a_{22}^k \mathbf{A}_k^T \mathbf{Y}_k \quad (3.4.29)$$

其中, $a_{11}^k, a_{12}^k, a_{21}^k$ 和 a_{22}^k 为任意参数。

式(3.4.28)和式(3.4.29)中的 \mathbf{U}_k 和 \mathbf{V}_k 已经满足式(3.4.20)和式(3.4.21)中 $j=0, 1, \dots, k-1$ 的情形, 而对于式(3.4.20)和式(3.4.21)中 $j=k$ 的情况, 只需要适当选择 $a_{11}^k, a_{12}^k, a_{21}^k$ 和 a_{22}^k 就能满足。

综上所述, 得到矩阵 \mathbf{A}_k 的递推公式为

$$\mathbf{A}_{k+1} = \mathbf{A}_k + \mathbf{S}_k \mathbf{U}_k^T + \mathbf{A}_k \mathbf{Y}_k \mathbf{V}_k^T \quad (3.4.30)$$

其中, \mathbf{U}_k 和 \mathbf{V}_k 由式(3.4.28)和式(3.4.29)给出, 且满足

$$\mathbf{U}_k^T \mathbf{Y}_k = \rho \quad (3.4.31)$$

$$\mathbf{V}_k^T \mathbf{Y}_k = -1 \quad (3.4.32)$$

式(3.4.28)~式(3.4.32)为一族变尺度法通用公式。 $\rho, a_{11}^k, a_{12}^k, a_{21}^k$ 和 a_{22}^k 的选择不同, 得到不同形式的 \mathbf{A}_k , 从而得到不同的变尺度法。

3.4.3 DFP 变尺度法

选择 $\rho=1$, $a_{12}^k=a_{21}^k=0$, 可推出递推矩阵

$$A_{k+1}=A_k+\frac{S_k S_k^T}{S_k^T Y_k}-\frac{A_k Y_k Y_k^T A_k}{Y_k^T A_k Y_k} \quad (3.4.33)$$

这就是 DFP (Davidon-Fletcher-Powell) 变尺度法的递推矩阵公式。

定理 3.4.1 若 A_0 为对称正定矩阵, 则由式 (3.4.33) 定义的 $A_k(k \geq 0)$ 亦为对称正定矩阵。

证 用归纳法证明, 由定理的条件 A_0 为对称正定矩阵。现假定 $A_i(i=1,2,\dots,k)$ 为对称正定矩阵。

下面, 证明 A_{k+1} 亦为对称正定矩阵。由于已经假定了 A_k 为对称正定矩阵, 显然, 由式 (3.4.33) 给出的 A_{k+1} 为对称矩阵。由于 A_k 为对称正定矩阵, 由式 (3.4.3) 得

$$(X^T A_k Y_k)^2 \leq X^T A_k X Y_k^T A_k Y_k \quad (3.4.34)$$

其中, $X \in R^n$ 。

$$\begin{aligned} S_k^T Y_k &= \lambda_k P_k^T [\nabla f(X_{k+1}) - \nabla f(X_k)] \\ &= -\lambda_k P_k^T \nabla f(X_k) \\ &= \lambda_k \nabla f(X_k)^T A_k \nabla f(X_k) > 0 \end{aligned} \quad (3.4.35)$$

设 $X \in R^n$, 且 $X \neq 0$, 有

$$\begin{aligned} X^T A_{k+1} X &= X^T A_k X + \frac{(X^T S_k)^2}{S_k^T Y_k} - \frac{(X^T A_k Y_k)^2}{Y_k^T A_k Y_k} \\ &= \frac{X^T A_k X Y_k^T A_k Y_k - (X^T A_k Y_k)^2}{Y_k^T A_k Y_k} + \frac{(X^T S_k)^2}{S_k^T Y_k} > 0 \end{aligned}$$

上式不等式的左边第一项大于等于 0, 当且仅当 $X = \lambda Y_k$ 时才能为 0; 当 $X = \lambda Y_k$ 时, 上式不等式左边第二项大于 0, 故定理成立, 证毕。

定理 3.4.2 设 A 为 $n \times n$ 阶对称正定矩阵, 考虑求解二次函数 $f(X) = \frac{1}{2} X^T A X + B^T X + c$

的极小点。取任意初始点 X_0 和初始对称正定矩阵 A_0 , 用 DFP 变尺度法求解, λ_i 是函数 $f(X_i + \lambda P_i)$ 的最优解, 令 $X_{i+1} = X_i + \lambda_i P_i$, 其中 $P_i = -A_i \nabla f(X_i)$, $A_i(i \geq 1)$ 由式 (3.4.33) 给出, 如果 $\nabla f(X_i) \neq 0$ ($0 \leq i \leq n-1$), 则

(1) $P_0, P_1, P_2, \dots, P_{n-1}$ 关于矩阵 A 共轭, 即

$$P_i^T A P_j = 0 \quad (0 \leq j < i \leq n-1) \quad (3.4.36)$$

(2) $A_{k+1} A P_i = P_i \quad (0 \leq i \leq k)$ (3.4.37)

(3) $A_n = A^{-1}$

证 首先, 用归纳法证明定理的结论 (1) 和结论 (2) 成立。为了证明方便, 令 $P_{n-1} = 0$ 向量。

由于 $f(X)$ 为二次函数和式 (3.4.33), 故有

$$Y_k = \nabla f(X_{k+1}) - \nabla f(X_k) = A S_k \quad (3.4.38)$$

$$\mathbf{A}_{k+1}\mathbf{A}\mathbf{P}_k = \frac{1}{\lambda_k}\mathbf{A}_{k+1}\mathbf{A}\mathbf{S}_k = \frac{1}{\lambda_k}\mathbf{A}_{k+1}\mathbf{Y}_k = \frac{1}{\lambda_k}\mathbf{S}_k = \mathbf{P}_k \quad (3.4.39)$$

由式 (3.4.39) 可知, 当 $k=0$ 时, 式 (3.4.36) 和式 (3.4.37) 成立。

现假设式 (3.4.36) 和式 (3.4.37) 对于 $k-1$ 成立, 即

$$\mathbf{P}_i^T \mathbf{A}\mathbf{P}_j = 0 \quad (0 \leq j < i \leq k-1) \quad (3.4.40)$$

$$\mathbf{A}_k \mathbf{A}\mathbf{P}_i = \mathbf{P}_i \quad (0 \leq i \leq k-1) \quad (3.4.41)$$

再证明对于 k , 式 (3.4.40) 和式 (3.4.41) 亦成立。由于函数 $f(\mathbf{X})$ 为二次函数, 故有

$$\nabla f(\mathbf{X}_k) = \nabla f(\mathbf{X}_{i+1}) + \sum_{j=i+1}^{k-1} \lambda_j \mathbf{A}\mathbf{P}_j$$

由于 $\mathbf{P}_i^T \nabla f(\mathbf{X}_{i+1}) = 0$, 由上式和式 (3.4.40), 得

$$\mathbf{P}_i^T \nabla f(\mathbf{X}_k) = 0 \quad (0 \leq i \leq k-1)$$

由上式和式 (3.4.41), 得

$$\mathbf{P}_i^T \mathbf{A}\mathbf{A}_k \nabla f(\mathbf{X}_k) = 0 \quad (0 \leq i \leq k-1)$$

由上式和 $\mathbf{P}_k = -\mathbf{A}_k \nabla f(\mathbf{X}_k)$, 有

$$\mathbf{P}_i^T \mathbf{A}\mathbf{P}_k = 0 \quad (0 \leq i \leq k-1) \quad (3.4.42)$$

式 (3.4.42) 说明定理的结论 (1) 成立。

另外, 由式 (3.4.41), 利用式 (3.4.38) 和式 (3.4.42), 可得

$$\mathbf{Y}_k^T \mathbf{A}_k \mathbf{A}\mathbf{P}_i = \mathbf{Y}_k^T \mathbf{P}_i = \lambda_k \mathbf{P}_k^T \mathbf{A}\mathbf{P}_i = 0 \quad (0 \leq i \leq k-1)$$

由上式、式 (3.4.33) 和式 (3.4.36), 有

$$\mathbf{A}_{k+1} \mathbf{A}\mathbf{P}_i = \mathbf{A}_k \mathbf{A}\mathbf{P}_i = \mathbf{P}_i \quad (0 \leq i \leq k-1)$$

上式与式 (3.4.39) 说明定理的结论 (2) 成立。

由定理 3.4.1 可知, $\mathbf{P}_i = -\mathbf{A}_i \nabla f(\mathbf{X}_i) \neq 0$ 向量, 由定理的结论 (1) 和共轭向量组的性质 1 可知 $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_{n-1}$ 线性无关, 令矩阵 $\mathbf{P} = [\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_{n-1}]$, 则 \mathbf{P} 为满秩矩阵, 由定理的结论 (2), 有

$$\mathbf{A}_n \mathbf{A}\mathbf{P} = \mathbf{P} \quad (3.4.43)$$

上式两边右乘 \mathbf{P}^{-1} , 得

$$\mathbf{A}_n \mathbf{A} = \mathbf{I} \quad (3.4.44)$$

由于 \mathbf{A}_n 和 \mathbf{A} 为对称矩阵, 由上式得

$$\mathbf{A}\mathbf{A}_n = \mathbf{I} \quad (3.4.45)$$

故 $\mathbf{A}_n = \mathbf{A}^{-1}$, 证毕。

我们要求 \mathbf{A}_{k+1} 为对称正定矩阵。但由于计算中的舍入误差, 特别是一维搜索不精确的影响, 可能在迭代过程中破坏 \mathbf{A}_{k+1} 正定性, 从而使计算失败。而验证矩阵的正定性的计算量很大。为了保证 \mathbf{A}_{k+1} 正定性, 我们采用如下两条措施。

(1) 一维搜索后, 如果迭代点的函数值不下降, 重置 \mathbf{A}_k 矩阵为单位矩阵 \mathbf{I} , 然后继续迭代。

(2) 迭代 $n+1$ 次后, 置 $\mathbf{X}_0 = \mathbf{X}_{n+1}$, $\mathbf{A}_0 = \mathbf{I}$, 重新迭代。

DFP 变尺度法的主要步骤如下:

(1) 给定初始点 \mathbf{X}_0 和收敛精度 ε 。

(2) 置 $\mathbf{A}_0 = \mathbf{I}$ (单位矩阵), $\mathbf{P}_0 = -\nabla f(\mathbf{X}_0)$, $k=0$ 。

(3) 设 \mathbf{X}_k 为第 k 次迭代点已经求得, 用一维搜索方法求 λ_k , 使 $f(\mathbf{X}_k + \lambda_k \mathbf{P}_k) = \min f(\mathbf{X}_k + \lambda \mathbf{P}_k)$, 置 $\mathbf{X}_{k+1} = \mathbf{X}_k + \lambda_k \mathbf{P}_k$ 。

(4) 若 $\|\nabla f(\mathbf{X}_{k+1})\| < \varepsilon$, 置 $\mathbf{X}^* = \mathbf{X}_{k+1}$, 并打印最优解 \mathbf{X}^* 和 $f(\mathbf{X}^*)$, 停止程序执行; 否则, 转到步骤 (5) [此处的范数为 $\|\mathbf{X}\| = (\mathbf{X}^T \mathbf{X})^{1/2}$]。

(5) 若 $f(\mathbf{X}_{k+1}) \geq f(\mathbf{X}_k)$, 置 $\mathbf{X}_0 = \mathbf{X}_k$, $\nabla f(\mathbf{X}_0) = \nabla f(\mathbf{X}_k)$, 转到步骤 (2); 否则, 转到步骤 (6)。

(6) 若 $k = n$, 置 $\mathbf{X}_0 = \mathbf{X}_{k+1}$, $\nabla f(\mathbf{X}_0) = \nabla f(\mathbf{X}_{k+1})$, 转到步骤 (2); 否则, 转到步骤 (7)。

(7) 计算 $\mathbf{Y}_k = \nabla f(\mathbf{X}_{k+1}) - \nabla f(\mathbf{X}_k)$, $\mathbf{S}_k = \mathbf{X}_{k+1} - \mathbf{X}_k$, $\mathbf{A}_{k+1} = \mathbf{A}_k + \frac{\mathbf{S}_k \mathbf{S}_k^T}{\mathbf{S}_k^T \mathbf{Y}_k} - \frac{\mathbf{A}_k \mathbf{Y}_k \mathbf{Y}_k^T \mathbf{A}_k}{\mathbf{Y}_k^T \mathbf{A}_k \mathbf{Y}_k}$, $\mathbf{P}_{k+1} = -\mathbf{A}_{k+1} \nabla f(\mathbf{X}_{k+1})$ 。

(8) $k = k + 1$, 转到步骤 (3)。

DFP 变尺度法程序框图如图 3.4.1 所示。

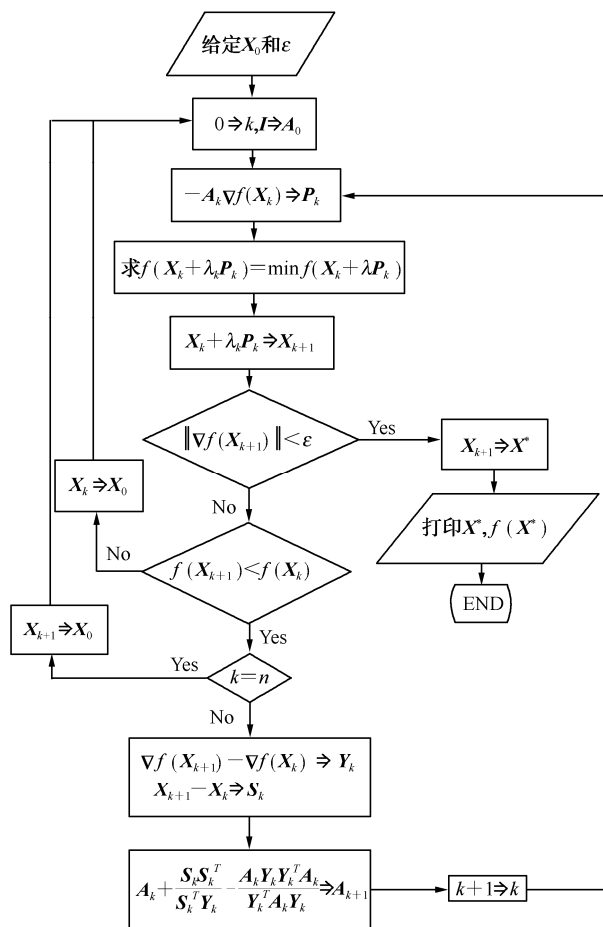


图 3.4.1 DFP 变尺度法程序框图

例 3.4.1 用变尺度法求函数 $f(\mathbf{X}) = 10(x_1 - x_2 - 5)^2 + (x_1 - x_2)^2$ 的极小点, 取初始点 $\mathbf{X}_0 = (0, 3)^T$, 本题的精确解为 $\mathbf{X}^* = (2.5, 2.5)^T$ 。

解 求函数 $f(\mathbf{X})=10(x_1+x_2-5)^2+(x_1-x_2)^2$ 在 \mathbf{X} 处的梯度

$$\nabla f(\mathbf{X})=\begin{pmatrix} 22x_1+18x_2-100 \\ 18x_1+22x_2-100 \end{pmatrix}$$

$$\nabla f(\mathbf{X}_0)=\begin{pmatrix} -46 \\ -34 \end{pmatrix}$$

令

$$\mathbf{A}_0=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

所以

$$\mathbf{P}_0=-\mathbf{A}_0\nabla f(\mathbf{X}_0)=\begin{pmatrix} 46 \\ 34 \end{pmatrix}$$

令

$$\mathbf{X}_0+\lambda\mathbf{P}_0=\begin{pmatrix} 0 \\ 3 \end{pmatrix}+\lambda\begin{pmatrix} 46 \\ 34 \end{pmatrix}=\begin{pmatrix} 46\lambda \\ 3+34\lambda \end{pmatrix}$$

$$\varphi(\lambda)=f(\mathbf{X}_0+\lambda\mathbf{P}_0)=10(80\lambda-2)^2+(12\lambda-3)^2$$

求函数 $\varphi(\lambda)$ 的驻点

$$\varphi'(\lambda_0)=128288\lambda_0-3272=0$$

由上式得

$$\lambda_0=0.0255051135$$

故有

$$\mathbf{X}_1=\mathbf{X}_0+\lambda_0\mathbf{P}_0=\begin{pmatrix} 46\lambda \\ 3+34\lambda \end{pmatrix}=\begin{pmatrix} 1.17323522 \\ 3.867173858 \end{pmatrix}$$

$$\nabla f(\mathbf{X}_1)=\begin{pmatrix} -4.57969572 \\ 6.19605883 \end{pmatrix}$$

$$\mathbf{Y}_0=\nabla f(\mathbf{X}_1)-\nabla f(\mathbf{X}_0)=\begin{pmatrix} 41.42030428 \\ 40.19605883 \end{pmatrix}$$

$$\mathbf{S}_0=\mathbf{X}_1-\mathbf{X}_0=\begin{pmatrix} 1.17323522 \\ 0.867173858 \end{pmatrix}$$

$$\mathbf{S}_0^T\mathbf{Y}_0=(1.17323522, 0.867173858)\begin{pmatrix} 41.42030428 \\ 40.19605883 \end{pmatrix}=83.45273121$$

$$\mathbf{Y}_0^T\mathbf{A}_0\mathbf{Y}_0=(41.42030428, 40.19605883)\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{pmatrix} 41.42030428 \\ 40.19605883 \end{pmatrix}=3331.364752$$

$$\begin{aligned}
A_1 &= A_0 + \frac{S_0 S_0^T}{S_0^T Y_0} - \frac{A_0 Y_0 Y_0^T A_0}{Y_0^T A_0 Y_0} \\
&= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{1}{83.45273121} \begin{pmatrix} 1.17323522 \\ 0.867173858 \end{pmatrix} (1.17323522, 0.867173858) \\
&\quad - \frac{1}{3331.364752} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} 41.42030428 \\ 40.19605883 \end{pmatrix} (41.42030428, 40.19605883) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 0.5014975 & -0.487583 \\ -0.487583 & 0.5240076 \end{bmatrix}
\end{aligned}$$

$$P_1 = -A_1 \nabla f(X_1) = - \begin{bmatrix} 0.5014975 & -0.487583 \\ -0.487583 & 0.5240076 \end{bmatrix} \begin{pmatrix} -4.57969572 \\ 6.19605883 \end{pmatrix} = \begin{pmatrix} 5.317798906 \\ -5.479763686 \end{pmatrix}$$

$$X_1 + \lambda P_1 = \begin{pmatrix} 1.17323522 \\ 3.867173858 \end{pmatrix} + \lambda \begin{pmatrix} 5.317798906 \\ -5.479763686 \end{pmatrix} = \begin{pmatrix} 1.17323522 + 5.317798906\lambda \\ 3.867173858 - 5.479763686\lambda \end{pmatrix}$$

$$\begin{aligned}
\varphi(\lambda) &= f(X_1 + \lambda P_1) = 10(-0.1619647792\lambda + 0.040409078)^2 \\
&\quad + (10.79756259\lambda - 2.693938638)^2
\end{aligned}$$

求函数 $\varphi(\lambda)$ 的驻点

$$\begin{aligned}
\varphi'(\lambda_1) &= 233.69936756\lambda_1 - 58.306839063 = 0 \\
\lambda_1 &= 0.2494950657
\end{aligned}$$

$$X_2 = X_1 + \lambda_1 P_1 = \begin{pmatrix} 1.17323522 + 5.317798906\lambda_1 \\ 3.867173858 - 5.479763686\lambda_1 \end{pmatrix} = \begin{pmatrix} 2.499999481 \\ 2.50000021 \end{pmatrix}$$

下面, 验证变尺度法的搜索方向 P_0 和 P_1 关于 A 共轭。

$$A = H(X) = \begin{pmatrix} 22 & 18 \\ 18 & 22 \end{pmatrix}$$

$$P_0^T A P_1 = (46 \quad 34) \begin{pmatrix} 22 & 18 \\ 18 & 22 \end{pmatrix} \begin{pmatrix} 5.317798906 \\ -5.479763686 \end{pmatrix} = -0.00214579 \approx 0$$

上式说明变尺度法的搜索方向 P_0 和 P_1 关于 A 共轭, 其误差是由于计算的舍入误差引起的。我们再验证 $A_2 = A^{-1}$ 。

$$\nabla f(X_2) = \begin{pmatrix} 0.0000076 \\ -0.0000047 \end{pmatrix} \approx \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$Y_1 = \nabla f(X_2) - \nabla f(X_1) = \begin{pmatrix} 4.57969572 \\ -6.19605883 \end{pmatrix}$$

$$S_1 = X_2 - X_1 = \begin{pmatrix} 1.32676426 \\ -1.367173648 \end{pmatrix}$$

$$S_1^T Y_1 = (1.32676426 \quad -1.367173648) \begin{pmatrix} 4.57969572 \\ -6.19605883 \end{pmatrix} = 14.547265$$

$$A_1 Y_1 = \begin{bmatrix} 0.5014975 & -0.487583 \\ -0.487583 & 0.5240076 \end{bmatrix} \begin{pmatrix} 4.57969572 \\ -6.19605883 \end{pmatrix} = \begin{pmatrix} 5.3177989 \\ -5.4797637 \end{pmatrix}$$

$$Y_1^T A_1 Y_1 = (4.57969572 \quad -6.19605883) \begin{pmatrix} 5.3177989 \\ -5.4797637 \end{pmatrix} = 58.306839$$

$$\begin{aligned} A_2 &= A_1 + \frac{S_1 S_1^T}{S_1^T Y_1} - \frac{A_1 Y_1 Y_1^T A_1}{Y_1^T A_1 Y_1} = \begin{bmatrix} 0.5014975 & -0.487583 \\ -0.487583 & 0.5240076 \end{bmatrix} \\ &\quad + \frac{1}{14.547265} \begin{pmatrix} 1.32676426 \\ -1.367173648 \end{pmatrix} (1.32676426 \quad -1.367173648) \\ &\quad - \frac{1}{58.306839} \begin{pmatrix} 5.3177989 \\ -5.4797637 \end{pmatrix} (5.3177989 \quad -5.4797637) \\ &= \begin{pmatrix} 0.137500 & -0.112500 \\ -0.112500 & 0.137500 \end{pmatrix} \end{aligned}$$

$$A^{-1} = \begin{pmatrix} 22 & 18 \\ 18 & 22 \end{pmatrix}^{-1} = \begin{pmatrix} 0.1375 & -0.1125 \\ -0.1125 & 0.1375 \end{pmatrix} = A_2$$

上式说明 $A_2 = A^{-1}$ 。

作者用抛物线插值法作为一维搜索方法，编写了变尺度法计算程序。用程序计算例 3.4.1 的有关数据如表 3.4.1 所示。

表 3.4.1 用变尺度法程序计算的有关数据

| 迭代次数 | X_k | P_k | $\nabla f(X_k)$ | A_k |
|------|--|---|---|--|
| 0 | $\begin{pmatrix} 0 \\ 3 \end{pmatrix}$ | $\begin{pmatrix} 46 \\ 34 \end{pmatrix}$ | $\begin{pmatrix} -46 \\ -34 \end{pmatrix}$ | $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ |
| 1 | $\begin{pmatrix} 1.173 \\ 3.867 \end{pmatrix}$ | $\begin{pmatrix} 5.318 \\ -5.480 \end{pmatrix}$ | $\begin{pmatrix} -4.580 \\ 6.196 \end{pmatrix}$ | $\begin{pmatrix} 0.507 & -0.488 \\ -0.488 & 0.524 \end{pmatrix}$ |
| 2 | $\begin{pmatrix} 2.5 \\ 2.5 \end{pmatrix}$ | | | |

例 3.4.2 用 DFP 变尺度法求函数 $f(\mathbf{X})=(x_1-2)^4+(x_1-2x_2)^2$ 的极小点, 取初始点 $\mathbf{X}_0=(0,3)^T$ 。用作者编写 DFP 变尺度法程序计算的最优点为 $\mathbf{X}^*=(2.039,1.020)^T$, $f(\mathbf{X}^*)=2.36 \times 10^{-6}$ 。[注: 精确解为 $\mathbf{X}^*=(2, 1)^T$, $f(\mathbf{X}^*)=0$], 计算的有关数据如表 3.4.2 所示, 搜索路线如图 3.4.2 所示。

表 3.4.2 变尺度法程序计算的有关数据

| 迭代次数 | \mathbf{X}_k | \mathbf{P}_k | $\nabla f(\mathbf{X}_k)$ | \mathbf{A}_k |
|------|--|---|---|--|
| 0 | $\begin{pmatrix} 0 \\ 3 \end{pmatrix}$ | $\begin{pmatrix} 44 \\ -24 \end{pmatrix}$ | $\begin{pmatrix} -44 \\ 24 \end{pmatrix}$ | $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ |
| 1 | $\begin{pmatrix} 2.708 \\ 1.523 \end{pmatrix}$ | $\begin{pmatrix} -0.697 \\ -1.377 \end{pmatrix}$ | $\begin{pmatrix} 0.739 \\ 1.355 \end{pmatrix}$ | $\begin{pmatrix} 0.251 & 0.377 \\ 0.377 & 0.810 \end{pmatrix}$ |
| 2 | $\begin{pmatrix} 2.554 \\ 1.219 \end{pmatrix}$ | $\begin{pmatrix} -0.072 \\ -0.007 \end{pmatrix}$ | $\begin{pmatrix} 0.909 \\ -0.460 \end{pmatrix}$ | $\begin{pmatrix} 0.128 & 0.097 \\ 0.097 & 0.810 \end{pmatrix}$ |
| 3 | $\begin{pmatrix} 2.316 \\ 1.197 \end{pmatrix}$ | $\begin{pmatrix} 0.029 \\ -0.312 \end{pmatrix}$ | $\begin{pmatrix} -0.029 \\ 0.312 \end{pmatrix}$ | $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ |
| 4 | $\begin{pmatrix} 2.319 \\ 1.161 \end{pmatrix}$ | $\begin{pmatrix} 0.103 \\ 0.053 \end{pmatrix}$ | $\begin{pmatrix} 0.125 \\ 0.012 \end{pmatrix}$ | $\begin{pmatrix} 0.792 & 0.396 \\ 0.396 & 0.323 \end{pmatrix}$ |
| 5 | $\begin{pmatrix} 2.040 \\ 1.018 \end{pmatrix}$ | $\begin{pmatrix} 9.7 \times 10^{-3} \\ -1.9 \times 10^{-2} \end{pmatrix}$ | $\begin{pmatrix} 9.7 \times 10^{-3} \\ -1.9 \times 10^{-2} \end{pmatrix}$ | $\begin{pmatrix} 2.147 & 1.074 \\ 1.074 & 0.661 \end{pmatrix}$ |
| 6 | $\begin{pmatrix} 2.039 \\ 1.020 \end{pmatrix}$ | | | |

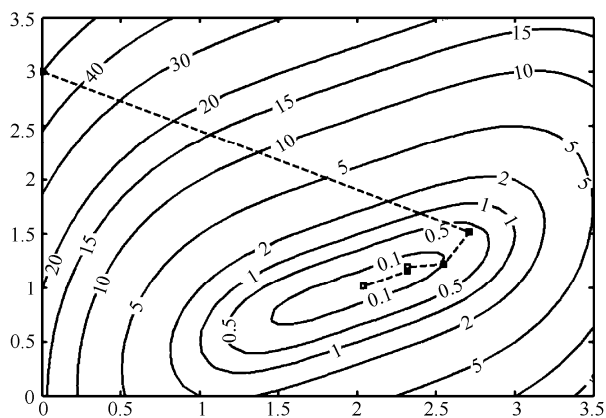


图 3.4.2 变尺度法的搜索路线图

3.4.4 BFGS 变尺度法

选择 $\rho=1$, $a_{12}^k=a_{21}^k=-\frac{1}{S_k^T Y_k}$, 可推出递推矩阵

$$A_{k+1}=A_k+\frac{(\beta_k S_k S_k^T - A_k Y_k S_k^T - S_k Y_k^T A_k)}{S_k^T Y_k} \quad (3.4.46)$$

式中, $\beta_k=1+\frac{Y_k^T A_k Y_k}{S_k^T Y_k}$ 。

BFGS (Broyden-Fletcher-Goldfarb-Shanno) 变尺度法与 DFP 变尺度法具有完全相同的性质, DFP 变尺度法的定理 3.4.1 和定理 3.4.2 对于 BFGS 变尺度法亦成立。而且, 由于 BFGS 变尺度法选择 $a_{12}^k=a_{21}^k \neq 0$, 使 A_k 不易成为病态矩阵, 故算法具有较好的数据稳定性, 这一点是 DFP 变尺度法无法比拟的。所以, BFGS 变尺度法是最成功的一种变尺度法。

3.5 模式搜索法

搜索方向最简单的取法是分别依次取各坐标方向, 这就是优化方法中的坐标轮换法。坐标轮换法程序简单, 但收敛速度慢。我们在这里不做介绍, 请读者参看文献[6]。下面我们来讨论由坐标轮换法产生出来的模式搜索法。

模式搜索法是由 Hooke-Jeeves 于 1961 年提出来的, 因此, 模式搜索法亦称为 Hooke-Jeeves 法。对于变量数目较少 (一般不超过 10 个变量) 的无约束最优化问题, 模式搜索法是一种程序简单、比较有效的优化方法。

算法由两部分组成, 第一部分是试探过程, 这个过程是确定有利的搜索方向; 第二部分是加速过程, 它是在有利方向上进行模式移动。该算法的主要步骤如下:

(1) 给定初始点 X_0 、初始搜索步长 h 和收敛精度 ε 。

(2) 从 X_0 出发, 依次沿各坐标方向

$$E_i = (0, \dots, 0, 1, 0, \dots, 0)^T \quad (i=1, 2, \dots, n)$$

以 h 为步长进行搜索, 取得下降迭代点

$$X_{0,i} = \begin{cases} X_{0,i-1} + hE_i & f(X_{0,i-1} + hE_i) < f(X_{0,i-1}) \\ X_{0,i-1} - hE_i & f(X_{0,i-1} - hE_i) < f(X_{0,i-1}) \\ X_{0,i-1} & f(X_{0,i-1}) \leq \min \{ f(X_{0,i-1} + hE_i), f(X_{0,i-1} - hE_i) \} \end{cases}$$

式中, $X_{0,0}=X_0$ 。经过这样的试探, 得到 $X_{0,n}$ 。

若 $X_{0,n} \neq X_{0,0}$, 转到步骤 (3); 若 $X_{0,n}=X_{0,0}$, 步长减半, 即置 $h=h/2$, 重新进行探索, 直到步长 h 缩小到足较小时, 仍得不到新的下降点, 则迭代停止, 置 $X^*=X_0$, 转到步骤 (6)。

(3) 进行模式移动。

令 $P_0=X_{0,n}-X_{0,0}$, 取 $\lambda_0=1$ (亦可进行一维搜索确定 λ_0), 可得

$$X_{0,n+1}=X_{0,n}+P_0=2X_{0,n}-X_{0,0}$$

若 $f(X_{0,n+1}) < f(X_{0,n})$, 置 $X_{1,0} = X_{0,n+1}$; 否则, 置 $X_{1,0} = X_{0,n}$ 。

(4) 假设已经求得了第 k 次迭代点 $X_{k,0}$, $k \geq 1$ 。从 $X_{k,0}$ 出发, 依次沿各坐标方向, 以 h 为步长进行搜索, 取得下降迭代点

$$X_{k,i} = \begin{cases} X_{k,i-1} + hE_i & f(X_{k,i-1} + hE_i) < f(X_{k,i-1}) \\ X_{k,i-1} - hE_i & f(X_{k,i-1} - hE_i) < f(X_{k,i-1}) \\ X_{k,i-1} & f(X_{k,i-1}) \leq \min\{f(X_{k,i-1} + hE_i), f(X_{k,i-1} - hE_i)\} \end{cases}$$

经过探索, 得到 $X_{k,n}$ 。

若 $X_{k,n} \neq X_{k,0}$, 转到步骤 (5); 若 $X_{k,n} = X_{k,0}$, 步长减半, 即置 $h = h/2$, 重新进行探索, 直到步长 $h < \varepsilon$, 仍得不到新的下降点, 则迭代停止, 置 $X^* = X_{k,0}$, 转到步骤 (6)。

(5) 进行模式移动。

令 $P_k = X_{k,n} - Z_{k-1,0}$, 取 $\lambda_k = 1$ (亦可进行一维搜索确定 λ_k), 可得

$$X_{k,n+1} = X_{k,n} + P_k = 2X_{k,n} - Z_{k-1,n}$$

若 $f(X_{k,n+1}) < f(X_{k,n})$, 置 $X_{k+1,0} = X_{k,n+1}$; 否则, 置 $X_{k+1,0} = X_{k,n}$, 转到步骤 (4)。

(6) 打印最优解 X^* 和 $f(X^*)$, 停止程序执行。

图 3.5.1 是二维模式搜索法的搜索轨迹。在 $X_{3,0}$ 处, 以原搜索步长 h 进行探索求不到新的下降点, 这时搜索步长 h 减半。

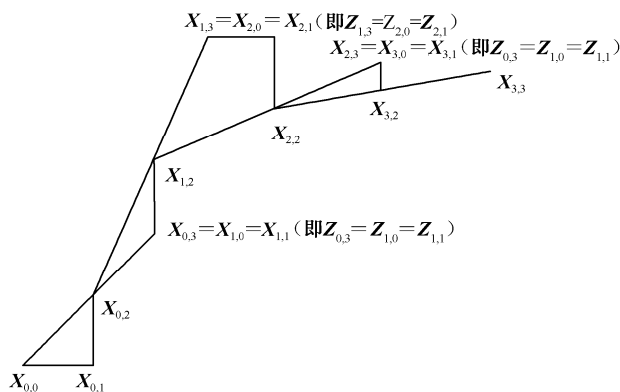


图 3.5.1 二维模式搜索法示意图

模式搜索法程序框图如图 3.5.2 所示。

例 3.5.1 用模式搜索求函数 $f(X) = x_1^2 + 2x_2^2 - 4x_1 - 2x_1x_2$ 的极小点, 取初始点 $X_0 = (0, 3)^T$, 初始步长 $h = 1$, 经过 39 次迭代, 用作者编写的模式搜索法程序的计算最优解为 $X^* = (3.998535, 1.999512)^T$, $f(X^*) = -8$ [注: 精确解为 $X^* = (4, 2)^T$, $f(X^*) = -8$], 前 10 步计算的搜索路线如图 3.5.3 所示, 计算数据如表 3.5.1 所示。

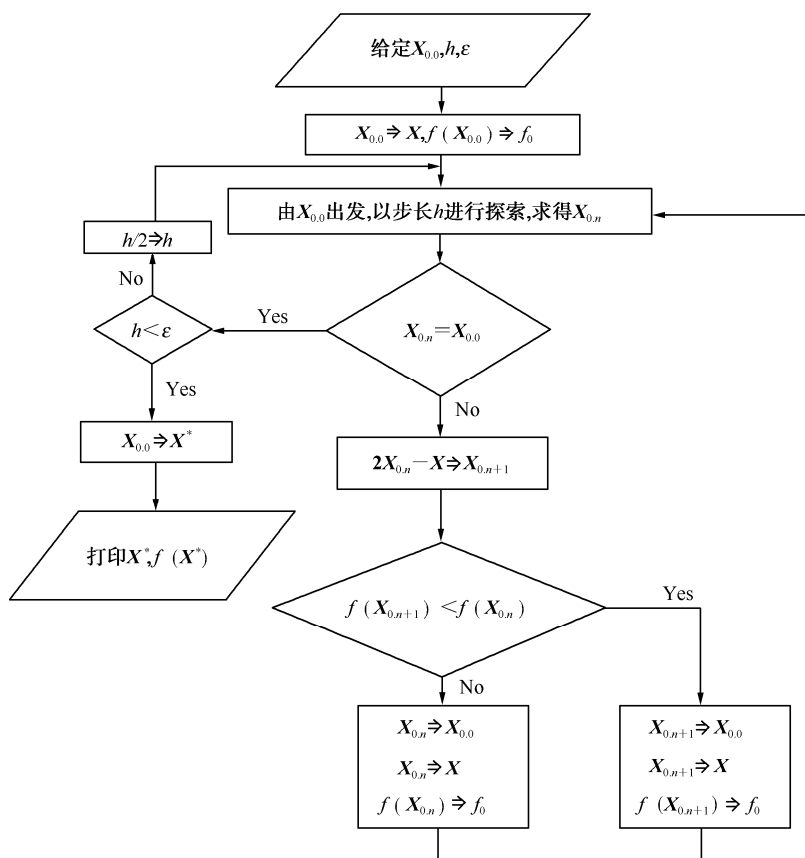


图 3.5.2 模式搜索法程序框图

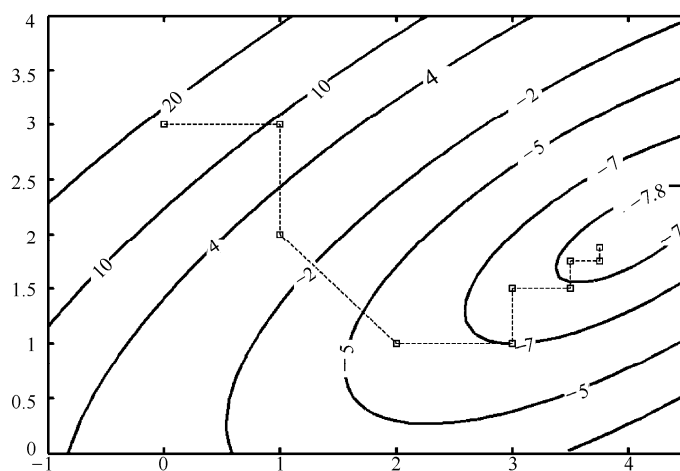


图 3.5.3 模式搜索法的搜索路线图

表 3.5.1 模式搜索法计算的有关数据

| k | h | $X_{k,0}$ | $f(X_{k,0})$ | $X_{k,1}$ | $f(X_{k,1})$ | $X_{k,2}$ | $f(X_{k,2})$ | 搜索 |
|-----|-------|--|--------------|--|--------------|---|--------------|----|
| 0 | 1 | $\begin{pmatrix} 0 \\ 3 \end{pmatrix}$ | 18 | $\begin{pmatrix} 1 \\ 3 \end{pmatrix}$ | 9 | $\begin{pmatrix} 1 \\ 2 \end{pmatrix}$ | 1 | 成功 |
| 1 | 1 | $\begin{pmatrix} 2 \\ 1 \end{pmatrix}$ | -6 | $\begin{pmatrix} 3 \\ 1 \end{pmatrix}$ | -7 | $\begin{pmatrix} 3 \\ 1 \end{pmatrix}$ | -7 | 成功 |
| 2 | 1 | $\begin{pmatrix} 3 \\ 1 \end{pmatrix}$ | -7 | $\begin{pmatrix} 3 \\ 1 \end{pmatrix}$ | -7 | $\begin{pmatrix} 3 \\ 1 \end{pmatrix}$ | -7 | 失败 |
| 3 | 0.5 | $\begin{pmatrix} 3 \\ 1 \end{pmatrix}$ | -7 | $\begin{pmatrix} 3 \\ 1 \end{pmatrix}$ | -7 | $\begin{pmatrix} 3 \\ 1.5 \end{pmatrix}$ | -7.5 | 成功 |
| 4 | 0.5 | $\begin{pmatrix} 3 \\ 1.5 \end{pmatrix}$ | -7.5 | $\begin{pmatrix} 3.5 \\ 1.5 \end{pmatrix}$ | -7.75 | $\begin{pmatrix} 3.5 \\ 1.5 \end{pmatrix}$ | -7.75 | 成功 |
| 5 | 0.5 | $\begin{pmatrix} 3.5 \\ 1.5 \end{pmatrix}$ | -7.75 | $\begin{pmatrix} 3.5 \\ 1.5 \end{pmatrix}$ | -7.75 | $\begin{pmatrix} 3.5 \\ 1.5 \end{pmatrix}$ | -7.75 | 失败 |
| 6 | 0.25 | $\begin{pmatrix} 3.5 \\ 1.5 \end{pmatrix}$ | -7.75 | $\begin{pmatrix} 3.5 \\ 1.5 \end{pmatrix}$ | -7.75 | $\begin{pmatrix} 3.5 \\ 1.75 \end{pmatrix}$ | -7.875 | 成功 |
| 7 | 0.25 | $\begin{pmatrix} 3.5 \\ 1.75 \end{pmatrix}$ | -7.875 | $\begin{pmatrix} 3.75 \\ 1.75 \end{pmatrix}$ | -7.9375 | $\begin{pmatrix} 3.75 \\ 1.75 \end{pmatrix}$ | -7.9375 | 成功 |
| 8 | 0.25 | $\begin{pmatrix} 3.75 \\ 1.75 \end{pmatrix}$ | -7.9375 | $\begin{pmatrix} 3.75 \\ 1.75 \end{pmatrix}$ | -7.9375 | $\begin{pmatrix} 3.75 \\ 1.75 \end{pmatrix}$ | -7.9375 | 失败 |
| 9 | 0.125 | $\begin{pmatrix} 3.75 \\ 1.75 \end{pmatrix}$ | -7.9375 | $\begin{pmatrix} 3.75 \\ 1.75 \end{pmatrix}$ | -7.9375 | $\begin{pmatrix} 3.75 \\ 1.875 \end{pmatrix}$ | -7.96875 | 成功 |

算法的几点说明:

(1) 算法所得结果的好坏与初始步长 h 的选择有关。在具体计算中, 可以对各变量采取不同的搜索步长 $h_i (i=1, 2, \dots, n)$ 。

(2) 模式移动方向亦可选用 $P_k = X_{k,n} - X_{k,0}$ 。

(3) 模式搜索法前面几步收敛较快, 但接近极小点时收敛速度慢。为了使算法的收敛速度有所改进, 模式搜索法可以与在极小点 X^* 附近收敛速度快的算法联用。

3.6 方向加速法

方向加速法是 1964 年由英国著名数学家 Powell^[7]提出的, 因此, 方向加速法又称为 Powell 法。在不依赖于目标函数梯度的所有直接法中, 方向加速法是最有效的方法。方向加速法的理论建立在二次函数的基础上, 它的基本理论是利用如下定理 3.6.1 逐次构成共轭方向, 并以此方向作为搜索方向, 因此, 从本质上讲方向加速法也是一种共轭方向法。

定理 3.6.1 设 n 维二次函数 $f(X) = \frac{1}{2} X^T A X + B^T X + c$, 其中 A 为 $n \times n$ 阶对称正定矩

阵, P_1, P_2, \dots, P_m 关于矩阵 A 共轭, 其中 $m < n$; X_1 和 X_2 是两个互异的点, 分别从 X_1 和 X_2 出

发, 依次沿 P_1, P_2, \dots, P_m 作一维搜索, 设 X_1^* 和 X_2^* 分别为它们最后一次一维搜索的极小点, 令 $P = X_1^* - X_2^*$, 则 P, P_1, P_2, \dots, P_m 关于矩阵 A 共轭。

证 设 $\lambda_1^k, \lambda_2^k, \dots, \lambda_m^k$ 分别为从 $X_k (k=1, 2)$ 出发, 沿 P_1, P_2, \dots, P_m 方向搜索的最优步长, 则有

$$\nabla f(X_k^*) = \nabla f(X_{i+1}) + \sum_{j=i+1}^{n-1} \lambda_j A P_j$$

由上式和定理的条件, 可得

$$P_i^T \nabla f(X_1^*) = 0 \quad (i=1, 2, \dots, m) \quad (3.6.1)$$

$$P_i^T \nabla f(X_2^*) = 0 \quad (i=1, 2, \dots, m) \quad (3.6.2)$$

上面两式相减, 得

$$P_i^T [\nabla f(X_1^*) - \nabla f(X_2^*)] = 0 \quad (i=1, 2, \dots, m) \quad (3.6.3)$$

由于 $f(X)$ 为二次函数, 故有

$$\nabla f(X_1^*) - \nabla f(X_2^*) = A(X_1^* - X_2^*) = AP \quad (3.6.4)$$

将上式代入式 (3.6.3), 得

$$P_i^T AP = 0 \quad (i=1, 2, \dots, m)$$

故 P, P_1, P_2, \dots, P_m 关于矩阵 A 共轭, 证毕。

下面, 用二维二次函数对定理 3.6.1 进行几何解释, 分别从 X_1 和 X_2 出发, 沿 P_1 方向进行一维搜索求得在 P_1 方向上的极小点分别为 X_1^* 和 X_2^* , 令 $P = X_1^* - X_2^*$, 则 P, P_1 关于二维二次函数 $f(X) = \frac{1}{2} X^T A X + B^T X + c$ 中的 A 矩阵共轭。 P 直指二维二次函数的极小点 X^* , 如图 3.6.1 所示。

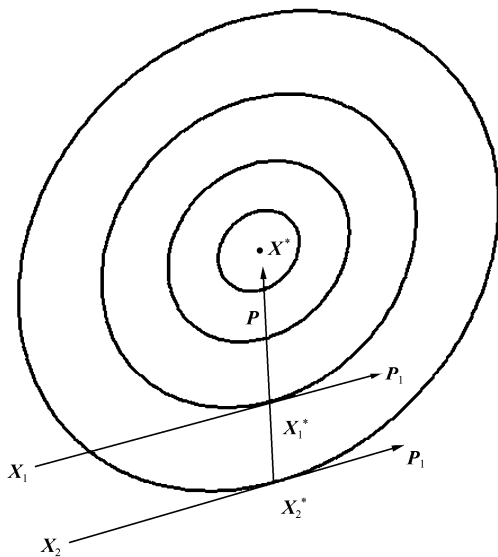


图 3.6.1 二维情形 Powell 法原理图

在 Powell 法中, 第一阶段迭代的出发点是选取的初始点, 第一阶段的搜索方向一般取各坐标轴的方向。而后, 每一个阶段迭代是从上一个阶段的迭代点出发, 沿 n 个线性无关的搜索方向分别作一维搜索。由这 n 次一维搜索求得的终点 $X_{k,n}$ 和出发点 $X_{k,0}$ 确定一个新的搜索

方向,用这个新的搜索方向代替原 n 个搜索方向中的一个搜索方向作为下一阶段的一组搜索方向。对于二次函数 $f(\mathbf{X}) = \frac{1}{2} \mathbf{X}^T \mathbf{A} \mathbf{X} + \mathbf{B}^T \mathbf{X} + c$, 经过 n 个阶段的迭代, 搜索方向组关于对称正定矩阵 \mathbf{A} 共轭。

原始 Powell 算法的主要步骤如下:

(1) 给定初始点 \mathbf{X}_0 、初始搜索方向 $\mathbf{P}_i = \mathbf{E}_i$ ($i=1, 2, \dots, n$), 其中, \mathbf{E}_i 是第 i 个坐标轴上的单位向量, 即

$$\mathbf{E}_i = (0, \dots, 0, \underset{i}{1}, 0, \dots, 0)^T \quad (i=1, 2, \dots, n)$$

(2) 依次沿 \mathbf{P}_i 方向进行一维搜索, 即求最优步长因子 λ_i , 使

$$f(\mathbf{X}_{i-1} + \lambda_i \mathbf{P}_i) = \min f(\mathbf{X}_{i-1} + \lambda \mathbf{P}_i)$$

令 $\mathbf{X}_i = \mathbf{X}_{i-1} + \lambda_i \mathbf{P}_i$ 。

(3) 置 $\mathbf{P}_{n+1} = \mathbf{X}_n - \mathbf{X}_0$ 。

(4) 从 \mathbf{X}_n 出发, 沿 \mathbf{P}_{n+1} 方向作一维搜索, 即使 $f(\mathbf{X}_n + \lambda_{n+1} \mathbf{P}_{n+1}) = \min f(\mathbf{X}_n + \lambda \mathbf{P}_{n+1})$, 令 $\mathbf{X}_{n+1} = \mathbf{X}_n + \lambda_{n+1} \mathbf{P}_{n+1}$ 。

(5) 若 $\|\mathbf{X}_{n+1} - \mathbf{X}_0\| < \varepsilon$, 令 $\mathbf{X}^* = \mathbf{X}_{n+1}$, 打印最优解 \mathbf{X}^* 和 $f(\mathbf{X}^*)$, 停止程序执行; 否则转到步骤 (6)。

(6) 置 $\mathbf{X}_0 = \mathbf{X}_{n+1}$ 和 $\mathbf{P}_i = \mathbf{P}_{i+1}$ ($i=1, 2, \dots, n$), 转到步骤 (2)。

原始 Powell 算法的缺点是在迭代阶段中往往使搜索方向组线性相关或近似线性相关, 事实上, 若在某一阶段的迭代过程中, 当 $\lambda_1 = 0$ (或 $\lambda_1 \approx 0$) 时, 则有

$$\mathbf{P}_{n+1} = \mathbf{X}_n - \mathbf{X}_0 = \sum_{i=2}^n \lambda_i \mathbf{P}_i$$

这时, \mathbf{P}_{n+1} 是 $\mathbf{P}_2, \mathbf{P}_3, \dots, \mathbf{P}_n$ 的线性组合。因此, 下一阶段的搜索方向 $\mathbf{P}_i = \mathbf{P}_{i+1}$ ($i=1, 2, \dots, n$) 是线性相关的, 致使迭代搜索求出来的极小点不是最优问题的极小点。我们用二维问题来说明这种情形。当从 \mathbf{X}_0 出发, 沿 $\mathbf{P}_1 = (1, 0)^T$ 方向进行一维搜索, 得 $\mathbf{X}_1 = \mathbf{X}_0$, 再从 \mathbf{X}_1 出发, 沿 $\mathbf{P}_2 = (0, 1)^T$ 方向进行一维搜索, 得 \mathbf{X}_2 。由于 $\lambda_1 = 0$, $\mathbf{P}_3 = \mathbf{X}_2 - \mathbf{X}_0 = \lambda_2 \mathbf{P}_2$, 因此, \mathbf{P}_2 与 \mathbf{P}_3 线性相关。由 \mathbf{X}_2 出发, 沿 \mathbf{P}_3 方向进行一维搜索, 得 $\mathbf{X}_3 = \mathbf{X}_2$ 。按原始 Powell 法, 下一阶段的搜索方向为 \mathbf{P}_2 和 \mathbf{P}_3 , 就求不到函数的极小点, 如图 3.6.2 所示。

现在有这样的一个问题, 新的搜索方向 $\mathbf{P}_{n+1} = \mathbf{X}_n - \mathbf{X}_0$ 是否被接纳作为下一阶段的搜索方向; 如果新的搜索方向 \mathbf{P}_{n+1} 被接纳的话, 去掉原搜索方向组中的哪一个搜索方向, 才能避免搜索方向组线性相关, 并能逐步地使搜索方向组变为共轭方向组。下面, 我们来讨论这个问题。只注重实用的读者, 可以跳过下面理论部分的论述, 直接阅读修改后的 Powell 法的主要步骤。

定理 3.6.2 设 \mathbf{A} 为 $n \times n$ 阶对称正定矩阵, 向量组 $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n$ 按下式规格化:

$$\mathbf{P}_i^T \mathbf{A} \mathbf{P}_i = 1 \quad (i=1, 2, \dots, n) \quad (3.6.5)$$

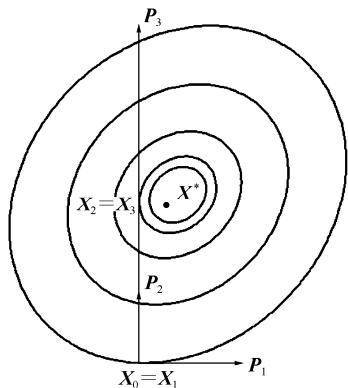


图 3.6.2 原始 Powell 法无法求得二维优化问题的极小点情形

令矩阵 $B=[P_1, P_2, \dots, P_n]$, 则 $|\det B|$ 为最大值的充要条件是 P_1, P_2, \dots, P_n 关于矩阵 A 共轭, 即

$$P_i^T A P_j = 0 \quad (i \neq j, 1 \leq i, j \leq n) \quad (3.6.6)$$

证 用反证法证明其必要性, 假设式 (3.6.6) 不成立, 不妨假设

$$P_1^T A P_2 = \beta \neq 0$$

由式 (3.4.3), 有

$$|P_1^T A P_2| \leq \|P_1^T A P_1\|^{1/2} \|P_2^T A P_2\|^{1/2} = 1$$

分两种情况讨论:

(1) 若 $|\beta| = 1$, 有 $P_1 = \pm P_2$, 根据行列式的性质, 有 $\det B = 0$ 。显然, $|\det B|$ 没有达到最大值。

(2) 若 $0 < |\beta| < 1$, 构造新的向量

$$V = \frac{P_1 - \beta P_2}{\sqrt{1 - \beta^2}}$$

容易证明

$$V^T A V = 1$$

令矩阵 $\bar{B}=[V, P_2, \dots, P_n]$, 有

$$\begin{aligned} |\det \bar{B}| &= \frac{1}{\sqrt{1 - \beta^2}} |\det(P_1 - \beta P_2, P_2, \dots, P_n)| \\ &= \frac{1}{\sqrt{1 - \beta^2}} |\det B| > |\det B| \end{aligned}$$

上式说明 $|\det B|$ 仍然没有达到最大值。

情况 (1) 和情况 (2) 证明了定理 3.6.2 的必要性。下面, 再来证明充分性。

根据必要性的结论, 按 A 矩阵规格化的向量组 Q_1, Q_2, \dots, Q_n 不是关于 A 矩阵共轭的一组向量, 则 $|\det(Q_1, Q_2, \dots, Q_n)|$ 不可能达到最大。因此, 要证明其充分性, 只需证明按 A 矩阵规格化的、关于 A 矩阵共轭的任意两组向量组 P_1, P_2, \dots, P_n 和 $\bar{P}_1, \bar{P}_2, \dots, \bar{P}_n$, 下式成立:

$$|\det B| = |\det \bar{B}|$$

其中, $B=[P_1, P_2, \dots, P_n]$ 和 $\bar{B}=[\bar{P}_1, \bar{P}_2, \dots, \bar{P}_n]$ 。

由于 P_1, P_2, \dots, P_n 关于对称正定矩阵 A 共轭, 由共轭向量组的性质 1 可知 P_1, P_2, \dots, P_n 线性无关, P_1, P_2, \dots, P_n 可作为 R^n 空间中的一组基向量。因此, $\bar{P}_1, \bar{P}_2, \dots, \bar{P}_n$ 可用 P_1, P_2, \dots, P_n 线性表示, 即

$$\bar{P}_j = \sum_{i=1}^n d_{ij} P_i$$

令 $D_j = (d_{1j}, d_{2j}, \dots, d_{nj})^T$, 则上式可表示为

$$\bar{P}_j = B D_j \quad (j=1, 2, \dots, n) \quad (3.6.7)$$

$$\bar{B} = B D \quad (3.6.8)$$

其中, $D=[D_1, D_2, \dots, D_n]$ 。

因为矩阵乘积的行列式等于各矩阵行列式的乘积, 故有

$$\det \bar{\mathbf{B}} = \det \mathbf{B} \det \mathbf{D} \quad (3.6.9)$$

由式 (3.6.8), 有

$$\bar{\mathbf{B}}^T \mathbf{A} \bar{\mathbf{B}} = \mathbf{D}^T \mathbf{B}^T \mathbf{A} \mathbf{B} \mathbf{D} = \mathbf{D}^T \mathbf{D} = \mathbf{I} \quad (3.6.10)$$

其中, \mathbf{I} 为 $n \times n$ 阶单位矩阵。

由式 (3.6.10), 可得

$$\det \mathbf{D} = \pm 1$$

将上式代入式 (3.6.9), 得 $|\det \mathbf{B}| = |\det \bar{\mathbf{B}}|$, 证毕。

定理 3.6.3 使用原始 Powell 法求 n 维二次函数 $f(\mathbf{X}) = \frac{1}{2} \mathbf{X}^T \mathbf{A} \mathbf{X} + \mathbf{B}^T \mathbf{X} + c$ 的极小点, 其中 \mathbf{A} 为 $n \times n$ 阶对称正定矩阵, 设 $\lambda_i (i=1, 2, \dots, n)$ 为从 \mathbf{X}_{i-1} 出发、沿 \mathbf{P}_i 方向的最优步长因子, $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n$ 和 $\mathbf{P}_{n+1} = \mathbf{X}_n - \mathbf{X}_0$ 都按下式进行规格化, 即选择正常数 t_i , 使

$$\mathbf{d}_i = \frac{1}{t_i} \mathbf{P}_i \quad (i=1, 2, \dots, n+1) \quad (3.6.11)$$

$$\mathbf{d}_i^T \mathbf{A} \mathbf{d}_i = 1 \quad (i=1, 2, \dots, n+1) \quad (3.6.12)$$

令

$$\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n]$$

和

$$\mathbf{D}_m = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_{m-1}, \mathbf{d}_{n+1}, \mathbf{d}_{m+1}, \dots, \mathbf{d}_n]$$

其中, $1 \leq m \leq n$ 。则有

$$\det \mathbf{D}_m = \frac{\lambda_m t_m}{t_{n+1}} \det \mathbf{D} \quad (3.6.13)$$

证 由定理的条件, 有

$$f(\mathbf{X}_{i-1} + \lambda_i \mathbf{P}_i) = \min f(\mathbf{X}_{i-1} + \lambda \mathbf{P}_i)$$

和

$$\mathbf{X}_i = \mathbf{X}_{i-1} + \lambda_i \mathbf{P}_i$$

故有

$$\begin{aligned} \mathbf{P}_{n+1} &= \mathbf{X}_n - \mathbf{X}_0 = \sum_{i=1}^n (\mathbf{X}_i - \mathbf{X}_{i-1}) \\ &= \sum_{i=1}^n \lambda_i \mathbf{P}_i = \sum_{i=1}^n \lambda_i t_i \mathbf{d}_i \\ \mathbf{d}_{n+1} &= \frac{\mathbf{P}_{n+1}}{t_{n+1}} = \sum_{i=1}^n \frac{\lambda_i t_i}{t_{n+1}} \mathbf{d}_i \end{aligned}$$

根据行列式的某一列乘以一个常数加到另一列上, 行列式的值不变。故有

$$\det \mathbf{D}_m = \det \left[\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_{m-1}, \sum_{i=1}^n \frac{\lambda_i t_i}{t_{n+1}} \mathbf{d}_i, \mathbf{d}_{m+1}, \dots, \mathbf{d}_n \right]$$

$$\begin{aligned}
&= \det \left[d_1, d_2, \dots, d_{m-1}, \frac{\lambda_m t_m}{t_{n+1}} d_m, d_{m+1}, \dots, d_n \right] \\
&= \det \left[d_1, d_2, \dots, d_{m-1}, \frac{\lambda_m t_m}{t_{n+1}} d_m, d_{m+1}, \dots, d_n \right] \\
&= \frac{\lambda_m t_m}{t_{n+1}} \det \mathbf{D}
\end{aligned}$$

证毕。

定理 3.6.4 在定理 3.6.3 的条件下, $|\lambda_i t_i|$ 、 $f(\mathbf{X}_{i-1}) - f(\mathbf{X}_i)$ 和 $|\det \mathbf{D}_i|$ ($i=1,2,\dots,n$) 这三组数据取得最大值, 必有相同的下标。

证 由式 (3.6.13) 可知, 则对于 $i \in (1,2,\dots,n)$, 下式成立:

$$|\det \mathbf{D}_i| = \left| \frac{\det \mathbf{D}}{t_{n+1}} \right| |\lambda_i t_i|$$

由于 $\left| \frac{\det \mathbf{D}}{t_{n+1}} \right|$ 为常数, 所以, $|\det \mathbf{D}_i|$ 和 $|\lambda_i t_i|$ 取得最大值, 必有相同的下标。

另外, 由于函数为二次函数, 以及 $\mathbf{X}_i = \mathbf{X}_{i-1} + \lambda_i \mathbf{P}_i$, 故有

$$\begin{aligned}
f(\mathbf{X}_{i-1}) - f(\mathbf{X}_i) &= \frac{1}{2} \mathbf{X}_{i-1}^T \mathbf{A} \mathbf{X}_{i-1} - \frac{1}{2} \mathbf{X}_i^T \mathbf{A} \mathbf{X}_i + \mathbf{B}^T (\mathbf{X}_{i-1} - \mathbf{X}_i) \\
&= \frac{1}{2} (\mathbf{X}_i - \lambda_i \mathbf{P}_i)^T \mathbf{A} (\mathbf{X}_i - \lambda_i \mathbf{P}_i) - \frac{1}{2} \mathbf{X}_i^T \mathbf{A} \mathbf{X}_i - \lambda_i \mathbf{B}^T \mathbf{P}_i \\
&= \frac{1}{2} \lambda_i^2 \mathbf{P}_i^T \mathbf{A} \mathbf{P}_i - \lambda_i \mathbf{P}_i^T \mathbf{A} \mathbf{X}_i - \lambda_i \mathbf{B}^T \mathbf{P}_i \\
&= \frac{1}{2} \lambda_i^2 \mathbf{P}_i^T \mathbf{A} \mathbf{P}_i - \lambda_i \mathbf{P}_i^T (\mathbf{A} \mathbf{X}_i + \mathbf{B})
\end{aligned} \tag{3.6.14}$$

由于 λ_i 是用一维搜索求得的最优步长因子, 即 $f(\mathbf{X}_{i-1} + \lambda_i \mathbf{P}_i) = \min f(\mathbf{X}_{i-1} + \lambda \mathbf{P}_i)$, 故有

$$\nabla f(\mathbf{X}_{i-1} + \lambda_i \mathbf{P}_i)^T \mathbf{P}_i = \nabla f(\mathbf{X}_i)^T \mathbf{P}_i = \mathbf{P}_i^T (\mathbf{A} \mathbf{X}_i + \mathbf{B}) = 0$$

而 $\mathbf{P}_i^T \mathbf{A} \mathbf{P}_i = t_i^2 d_i^T \mathbf{A} d_i = t_i^2$, 将此式和上式代入式 (3.6.14), 得

$$f(\mathbf{X}_{i-1}) - f(\mathbf{X}_i) = \frac{1}{2} \lambda_i^2 t_i^2 \tag{3.6.15}$$

式 (3.6.15) 说明 $f(\mathbf{X}_{i-1}) - f(\mathbf{X}_i)$ 和 $|\lambda_i t_i|$ 取最大值的下标相同, 故定理成立, 证毕。

由定理 3.6.3 得知, 当 $\left| \frac{\lambda_m t_m}{t_{n+1}} \right| > 1$ 时, 用 \mathbf{P}_{n+1} 代替原搜索方向组中的 \mathbf{P}_m 。很自然地取 m 使

$|\lambda_m t_m| \geq |\lambda_i t_i|$ ($i=1,2,\dots,n$)。当 $\max \left| \frac{\lambda_m t_m}{t_{n+1}} \right| \leq 1$, 保持原搜索方向组不变。而定理 3.6.4 提供

了一个简单的求得 m 的方法, 即根据迭代点的函数值 $f(\mathbf{X}_{i-1}) - f(\mathbf{X}_i)$ 的大小就能确定 m 。

现在的问题是如何用较简单的方法来判断 $\left| \frac{\lambda_m t_m}{t_{n+1}} \right|$ 是否大于 1。下面, 我们将讨论这个问题。

为了简化记号, 令 $\Delta_m = \max \{f(\mathbf{X}_{i-1}) - f(\mathbf{X}_i), i=1, 2, \dots, n\}$, $f_1=f(\mathbf{X}_0)$, $f_2=f(\mathbf{X}_n)$, $f_3=f(2\mathbf{X}_n - \mathbf{X}_0)$, $\bar{f}=f(\mathbf{X}_n + \bar{\lambda}\mathbf{P}_{n+1})$, 其中 $\bar{\lambda}$ 为从 \mathbf{X}_n 出发、沿 \mathbf{P}_{n+1} 方向上作一维搜索的最优步长因子。

考虑搜索方向 $\mathbf{P}_{n+1} = \mathbf{X}_n - \mathbf{X}_0$ 上等距的三个点 \mathbf{X}_0 、 \mathbf{X}_n 和 $2\mathbf{X}_n - \mathbf{X}_0$, 由于函数为二次函数, 即 $f(\mathbf{X}) = \frac{1}{2}\mathbf{X}^T \mathbf{A}\mathbf{X} + \mathbf{B}^T \mathbf{X} + c$, 其中 \mathbf{A} 为 $n \times n$ 阶对称正定矩阵, 因此, 经过 \mathbf{X}_n 点沿 \mathbf{P}_{n+1} 方向, 函数为抛物线。设 $\bar{\mathbf{X}}$ 为抛物线的极小点, 则

$$\bar{\mathbf{X}} = \mathbf{X}_n - \frac{f_3 - f_1}{2(f_1 - 2f_2 + f_3)} \mathbf{P}_{n+1} \quad (3.6.16)$$

和

$$\bar{f} = f_2 - \frac{(f_3 - f_1)^2}{8(f_1 - 2f_2 + f_3)} \quad (3.6.17)$$

$$\text{证 令 } \varphi(\lambda) = f(\mathbf{X}_n + \lambda\mathbf{P}_{n+1}) = a_0 + a_1(\lambda+1) + a_2\lambda(\lambda+1) \quad (3.6.18)$$

当 $\lambda = -1$ 时, 有

$$f_1 = a_0 \quad (3.6.19)$$

当 $\lambda = 0$ 时, 有

$$\begin{aligned} f_2 &= f_1 + a_1 \\ a_1 &= f_2 - f_1 \end{aligned} \quad (3.6.20)$$

当 $\lambda = 1$ 时, 有

$$\begin{aligned} f_3 &= f_1 + 2(f_2 - f_1) + 2a_2 \\ a_2 &= \frac{(f_1 - 2f_2 + f_3)}{2} \end{aligned} \quad (3.6.21)$$

令 $\varphi'(\bar{\lambda}) = a_1 + a_2(2\bar{\lambda} + 1) = 0$, 有

$$\bar{\lambda} = -\frac{a_1 + a_2}{2a_2} = -\frac{f_3 - f_1}{2(f_1 - 2f_2 + f_3)} \quad (3.6.22)$$

$$\bar{\mathbf{X}} = \mathbf{X}_n + \bar{\lambda}\mathbf{P}_{n+1} = \mathbf{X}_n - \frac{f_3 - f_1}{2(f_1 - 2f_2 + f_3)} \mathbf{P}_{n+1}$$

将式 (3.6.19) ~ 式 (3.6.22) 代入式 (3.6.18), 得

$$\bar{f} = f_2 - \frac{(f_3 - f_1)^2}{8(f_1 - 2f_2 + f_3)}$$

证毕。

由式 (3.6.15), 得

$$\lambda_m t_m = \pm \sqrt{2[f(X_{m-1}) - f(X_m)]} = \pm \sqrt{2\Delta_m} \quad (3.6.23)$$

设 $\bar{X} = X_0 + \bar{\lambda}_1 P_{n+1}$ 和 $\bar{X} = X_n + \bar{\lambda}_2 P_{n+1}$, 有

$$P_{n+1} = X_n - X_0 = (\bar{\lambda}_1 - \bar{\lambda}_2) P_{n+1} \quad (3.6.24)$$

因此,

$$\bar{\lambda}_1 - \bar{\lambda}_2 = 1.$$

用推导式 (3.6.15) 的方法, 类似地可得

$$f_1 - \bar{f} = \frac{1}{2} \bar{\lambda}_1^2 t_{n+1}^2$$

由于 $\bar{\lambda}_1 > 0$, 故有

$$\bar{\lambda}_1 t_{n+1} = \sqrt{2(f_1 - \bar{f})} \quad (3.6.25)$$

同理可得

$$\bar{\lambda}_2 t_{n+1} = \pm \sqrt{2(f_2 - \bar{f})} \quad (3.6.26)$$

由式 (3.6.24) ~ 式 (3.6.26), 可得

$$t_{n+1} = \sqrt{2(f_1 - \bar{f})} \pm \sqrt{2(f_2 - \bar{f})} \quad (3.6.27)$$

当 $f_3 \geq f_1$ 时, 由式 (3.6.16) 可知, \bar{X} 必须在 X_0 和 X_n 之间, 这时 $\bar{\lambda}_2 < 0$, 式 (3.6.26) 应取负号, 有

$$t_{n+1} = \sqrt{2(f_1 - \bar{f})} + \sqrt{2(f_2 - \bar{f})} \quad (3.6.28)$$

因为 $f_1 > f_2 \geq \bar{f}$, 故有

$$\begin{aligned} t_{n+1} &= \sqrt{2(f_1 - \bar{f})} + \sqrt{2(f_2 - \bar{f})} \\ &\geq \sqrt{2(f_1 - \bar{f})} = \sqrt{2(f_1 - f_2) + 2(f_2 - \bar{f})} \\ &\geq \sqrt{2(f_1 - f_2)} \end{aligned} \quad (3.6.29)$$

而

$$\begin{aligned} 2(f_1 - f_2) &= 2[f(X_0) - f(X_n)] \\ &= 2 \sum_{i=1}^n [f(X_{i-1}) - f(X_i)] \\ &= \sum_{i=1}^n (\lambda_i t_i)^2 \geq (\lambda_n t_n)^2 \end{aligned} \quad (3.6.30)$$

由式 (3.6.29) 和式 (3.6.30), 可得 $\left| \frac{\lambda_n t_n}{t_{n+1}} \right| \leq 1$ 。

综上所述, 当 $f_3 \geq f_1$ 时, 有 $\left| \frac{\lambda_n t_n}{t_{n+1}} \right| \leq 1$, 因此, 下一阶段的搜索方向组保持原搜索方向组

不变。

当 $f_3 < f_1$ 时, \bar{X} 必须在点 X_n 和 $2X_n - X_0$ 之间, 这时 $\bar{\lambda}_2 > 0$, 式 (3.6.26) 应取正号, 有

$$t_{n+1} = \sqrt{2(f_1 - \bar{f})} - \sqrt{2(f_2 - \bar{f})} \quad (3.6.31)$$

为了使 $\left| \frac{\lambda_n t_n}{t_{n+1}} \right| > 1$, 则有

$$\lambda_m^2 t_m^2 > t_{n+1}^2 \quad (3.6.32)$$

将式 (3.6.31)、式 (3.6.23) 和式 (3.6.17) 代入式 (3.6.32), 可得

$$2\Delta_m > 2(f_1 - \bar{f}) + 2(f_2 - \bar{f}) - 4\sqrt{(f_1 - \bar{f})(f_2 - \bar{f})} \quad (3.6.33)$$

由上式, 可得

$$2\sqrt{(f_1 - \bar{f})(f_2 - \bar{f})} > f_1 + f_2 - 2\bar{f} - \Delta_m \quad (3.6.34)$$

由上式右边项 $f_1 + f_2 - 2\bar{f} - \Delta_m \geq 0$, 上式两边平方, 可得

$$4(f_1 - \bar{f})(f_2 - \bar{f}) > (f_1 + f_2 - 2\bar{f} - \Delta_m)^2 \quad (3.6.35)$$

而 $(f_1 + f_2 - 2\bar{f} - \Delta_m)^2 = (f_1 - f_2 - \Delta_m)^2 + 4(f_2 - \bar{f})(f_1 - f_2 - \Delta_m) + 4(f_2 - \bar{f})^2$

将上式代入式 (3.6.35), 可得

$$4(f_2 - \bar{f})\Delta_m > (f_1 - f_2 - \Delta_m)^2 \quad (3.6.36)$$

将式 (3.6.17) 代入式 (3.6.36), 可得

$$(f_1 - 2f_2 + f_3)(f_1 - f_2 - \Delta_m)^2 < 0.5\Delta_m(f_1 - f_3)^2 \quad (3.6.37)$$

上式等价于 $\left| \frac{\lambda_m t_m}{t_{n+1}} \right| > 1$ 。

综上所述, 只有

$$(1) f_3 < f_1 \quad (3.6.38)$$

$$(2) (f_1 - 2f_2 + f_3)(f_1 - f_2 - \Delta_m)^2 < 0.5\Delta_m(f_1 - f_3)^2 \quad (3.6.39)$$

同时成立时, 才用 \mathbf{P}_{n+1} 代替原搜索方向组中的 \mathbf{P}_m ; 否则保持原有搜索方向不变。

文献[8]中证明了式 (3.6.38) 和式 (3.6.39) 的等价公式为

$$\frac{(f_1 - 2f_2 + f_3)}{2} < \Delta_m \quad (3.6.40)$$

证 若式 (3.6.40) 成立, 则有

$$f_1 - f_2 - \Delta_m < \frac{f_1 - f_3}{2} \quad (3.6.41)$$

又因为

$$f_1 \geq f_{m-1} \geq f_m \geq f_2 \quad (3.6.42)$$

其中, $f_{m-1} = f(X_{m-1})$, $f_m = f(X_m)$ 。

由式 (3.6.40) 和式 (3.6.41), 得

$$\frac{f_1 - f_3}{2} > f_1 - f_2 - \Delta_m = f_1 - f_2 - f_{m-1} + f_m \geq 0 \quad (3.6.43)$$

所以 $f_3 < f_1$, 即式 (3.6.38) 成立。

由式 (3.6.41)、式 (3.6.42) 和 $f_1 - f_2 - \Delta_m$ 的非负性, 可得

$$(f_1 - 2f_2 + f_3)(f_1 - f_2 - \Delta_m)^2 < 0.5\Delta_m(f_1 - f_3)^2$$

成立, 即式 (3.6.39) 成立。

现在, 用反证法来证明若式 (3.6.38) 和式 (3.6.39) 成立, 则式 (3.6.40) 成立。假设式 (3.6.40) 不成立, 即

$$\frac{(f_1 - 2f_2 + f_3)}{2} \geq \Delta_m \geq 0 \quad (3.6.44)$$

于是, 有

$$f_1 - f_2 - \Delta_m \geq \frac{f_1 - f_3}{2} > 0 \quad (3.6.45)$$

从而, 由式 (3.6.44) 和式 (3.6.45) 可得

$$(f_1 - 2f_2 + f_3)(f_1 - f_2 - \Delta_m)^2 \geq 0.5\Delta_m(f_1 - f_3)^2$$

上式与式 (3.6.39) 矛盾, 因此, 式 (3.6.40) 必须成立, 证毕。

改进的 Powell 法的主要步骤如下:

(1) 给定初始点 \mathbf{X}_0 、初始搜索方向 $\mathbf{P}_i = \mathbf{E}_i$ ($i=1, 2, \dots, n$), 其中 \mathbf{E}_i 是第 i 个坐标轴上的单位向量, 即

$$\mathbf{E}_i = (0, \dots, 0, \overset{i}{1}, 0, \dots, 0)^T \quad (i=1, 2, \dots, n)$$

(2) 依次沿 \mathbf{P}_i 方向进行一维搜索, 即求最优步长因子 λ_i , 使

$$f(\mathbf{X}_{i-1} + \lambda_i \mathbf{P}_i) = \min f(\mathbf{X}_{i-1} + \lambda \mathbf{P}_i)$$

令 $\mathbf{X}_i = \mathbf{X}_{i-1} + \lambda_i \mathbf{P}_i$ 。

(3) 置 $\mathbf{P}_{n+1} = \mathbf{X}_n - \mathbf{X}_0$ 。

(4) 置 $f_1 = f(\mathbf{X}_0)$ 、 $f_2 = f(\mathbf{X}_n)$ 、 $f_3 = f(2\mathbf{X}_n - \mathbf{X}_0)$ 和 $\Delta_m = \max \{f(\mathbf{X}_{i-1}) - f(\mathbf{X}_i)\}$ 。

(5) 若 $\frac{f_1 - 2f_2 + f_3}{2} \geq \Delta_m$, 转到步骤 (8)。

(6) 求最优步长因子 λ_{n+1} , 使

$$f(\mathbf{X}_n + \lambda_{n+1} \mathbf{P}_{n+1}) = \min f(\mathbf{X}_n + \lambda \mathbf{P}_{n+1})$$

令 $\mathbf{X}_{n+1} = \mathbf{X}_n + \lambda_{n+1} \mathbf{P}_{n+1}$ 。

(7) 置 $\mathbf{P}_j = \mathbf{P}_j$ ($j=1, 2, \dots, m-1$) 和 $\mathbf{P}_j = \mathbf{P}_{j+1}$ ($j=m, m+1, \dots, n$), 转到步骤 (9)。

(8) 如果 $f_2 \leq f_3$, 置 $\mathbf{X}_{n+1} = \mathbf{X}_n$; 否则, $\mathbf{X}_{n+1} = 2\mathbf{X}_n - \mathbf{X}_0$ 。

(9) 若 $\|\mathbf{X}_{n+1} - \mathbf{X}_0\| < \varepsilon$, 令 $\mathbf{X}^* = \mathbf{X}_{n+1}$, 打印最优解 \mathbf{X}^* 和 $f(\mathbf{X}^*)$, 停止程序执行; 否则, 置 $\mathbf{X}_0 = \mathbf{X}_{n+1}$, 转到步骤 (2)。

改进的 Powell 法程序框图如图 3.6.3 所示。

例 3.6.1 用修改的 Powell 法求函数 $f(\mathbf{X}) = x_1^2 + 2x_2^2 - 4x_1 - 2x_1x_2$ 的极小点, 取初始点 $\mathbf{X}_0 = (0, 3)^T$, 取初始搜索方向为坐标方向, 即 $\mathbf{P}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ 和 $\mathbf{P}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ 。

解 $f_1 = f(\mathbf{X}_0) = 18$, 令 $\mathbf{X} = \mathbf{X}_0 + \lambda \mathbf{P}_1 = \begin{pmatrix} 0 \\ 3 \end{pmatrix} + \lambda \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \lambda \\ 3 \end{pmatrix}$, 则有

$$\varphi(\lambda) = f(\mathbf{X}_0 + \lambda \mathbf{P}_1) = \lambda^2 - 10\lambda + 18$$

求 $\min \varphi(\lambda)$, 可得

$$\lambda_1 = 5$$

$$\mathbf{X}_1 = \mathbf{X}_0 + \lambda_1 \mathbf{P}_1 = \begin{pmatrix} 0 \\ 3 \end{pmatrix} + 5 \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 5 \\ 3 \end{pmatrix}$$

$$f(\mathbf{X}_1) = -7$$

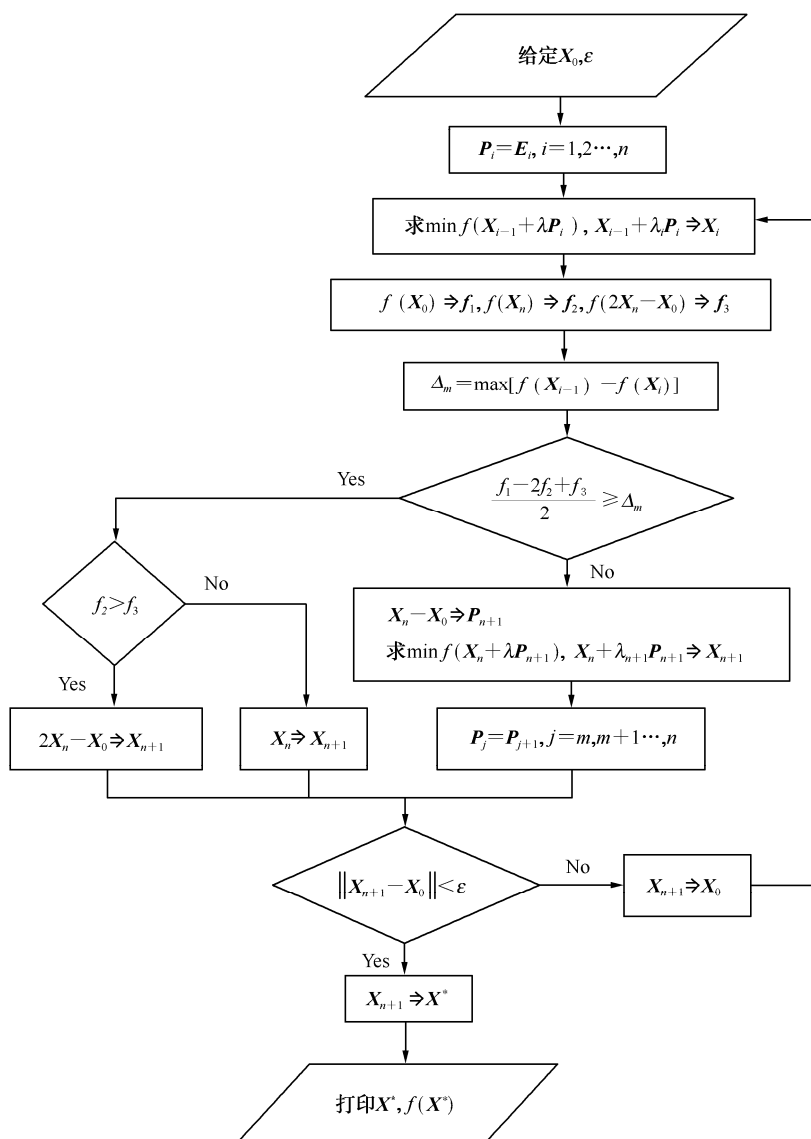


图 3.6.3 改进的 Powell 法程序框图

令 $X = X_1 + \lambda P_2 = \begin{pmatrix} 5 \\ 3 \end{pmatrix} + \lambda \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 5 \\ 3 + \lambda \end{pmatrix}$, 则有

$$\varphi(\lambda) = f(X_1 + \lambda P_2) = 2(3 + \lambda)^2 - 10(3 + \lambda) + 5$$

求 $\min \varphi(\lambda)$, 可得

$$\lambda_2 = -0.5$$

$$X_2 = X_1 + \lambda_2 P_2 = \begin{pmatrix} 5 \\ 3 \end{pmatrix} - 0.5 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 5 \\ 2.5 \end{pmatrix}$$

$$f_2=f(X_2)=-7.5$$

$$f_3=f(2X_2-X_0)=18$$

$$\Delta_m=f(X_0)-f(X_1)=25$$

将上面有关数据代入, 可得 $\frac{(f_1-2f_2+f_3)}{2} \geq \Delta_m$, 因此, 保持原搜索方向组不变, 置

$X_0=X_2=\begin{pmatrix} 5 \\ 2.5 \end{pmatrix}$, 进入第二阶段的迭代。

令 $X=X_0+\lambda P_1=\begin{pmatrix} 5 \\ 2.5 \end{pmatrix}+\lambda\begin{pmatrix} 1 \\ 0 \end{pmatrix}=\begin{pmatrix} 5+\lambda \\ 2.5 \end{pmatrix}$, $f_1=f(X_0)=-7.5$, 则有

$$\varphi(\lambda)=f(X_0+\lambda P_1)=(5+\lambda)^2-9(5+\lambda)+12.5$$

求 $\min \varphi(\lambda)$, 可得

$$\lambda_1=-0.5$$

$$X_1=X_0+\lambda_1 P_1=\begin{pmatrix} 5 \\ 2.5 \end{pmatrix}-0.5\begin{pmatrix} 1 \\ 0 \end{pmatrix}=\begin{pmatrix} 4.5 \\ 2.5 \end{pmatrix}$$

$$f(X_1)=-7.75$$

令 $X=X_1+\lambda P_2=\begin{pmatrix} 4.5 \\ 2.5 \end{pmatrix}+\lambda\begin{pmatrix} 0 \\ 1 \end{pmatrix}=\begin{pmatrix} 4.5 \\ 2.5+\lambda \end{pmatrix}$, 则有

$$\varphi(\lambda)=f(X_1+\lambda P_2)=2(2.5+\lambda)^2-9(2.5+\lambda)-2.25$$

求 $\min \varphi(\lambda)$, 可得

$$\lambda_2=-0.25$$

$$X_2=X_1+\lambda_2 P_2=\begin{pmatrix} 4.5 \\ 2.5 \end{pmatrix}-0.25\begin{pmatrix} 0 \\ 1 \end{pmatrix}=\begin{pmatrix} 4.5 \\ 2.25 \end{pmatrix}$$

$$f_2=f(X_2)=-7.875$$

$$f_3=f(2X_2-X_0)=-8$$

$$\Delta_m=f(X_0)-f(X_1)=0.25$$

将上面第二阶段有关数据代入, 可得 $\frac{(f_1-2f_2+f_3)}{2} < \Delta_m$, 因此, 用 $P_3=X_2-X_0=\begin{pmatrix} -0.5 \\ -0.25 \end{pmatrix}$

代替 P_1 。第三阶段的搜索方向组 $P_1=P_2=\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ 和 $P_2=P_3=\begin{pmatrix} -0.5 \\ -0.25 \end{pmatrix}$ 。

从 X_2 出发, 沿 $P_3=\begin{pmatrix} -0.5 \\ -0.25 \end{pmatrix}$ 作一维搜索, 即 $\varphi(\lambda)=f(X_2+\lambda_3 P_3)=\min f(X_2+\lambda P_3)$, 可得

$\lambda_3=1$; $\mathbf{X}_3=\mathbf{X}_2+\lambda_3\mathbf{P}_3=\begin{pmatrix} 4.5 \\ 2.25 \end{pmatrix}+1\begin{pmatrix} -0.5 \\ -0.25 \end{pmatrix}=\begin{pmatrix} 4 \\ 2 \end{pmatrix}$ 。因为,本例的函数为二次函数,且已经在共轭

方向组 $\mathbf{P}_2=\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ 和 $\mathbf{P}_3=\begin{pmatrix} -0.5 \\ -0.25 \end{pmatrix}$ 分别作了两次一维搜索,故 $\mathbf{X}^*=\mathbf{X}_3$ 。函数的等值线和用修改的

Powell 法求解搜索路线如图 3.6.4 所示。

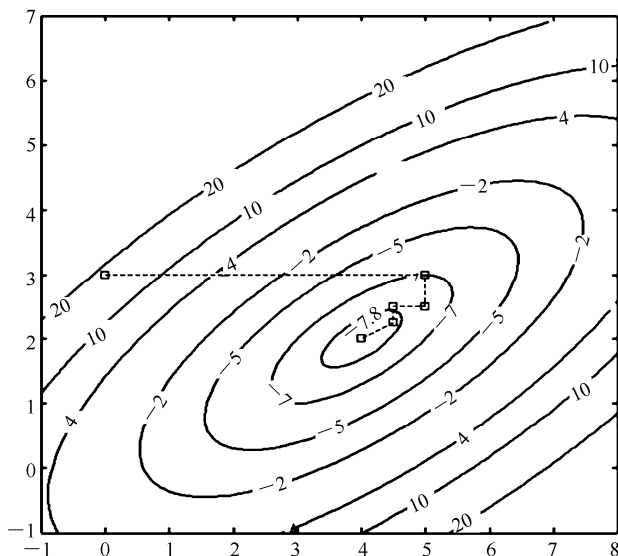


图 3.6.4 修改的 Powell 法的搜索路线图

下面,验证 $\mathbf{P}_2=\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ 和 $\mathbf{P}_3=\begin{pmatrix} -0.5 \\ -0.25 \end{pmatrix}$ 关于二次函数 $f(\mathbf{X})=x_1^2+2x_2^2-4x_1-2x_1x_2$ 的 Hessian

矩阵共轭。 $\mathbf{H}(\mathbf{X})=\begin{pmatrix} 2 & -2 \\ -2 & 4 \end{pmatrix}$, $\mathbf{P}_2^T\mathbf{H}(\mathbf{X})\mathbf{P}_3=(0 \ 1)\begin{pmatrix} 2 & -2 \\ -2 & 4 \end{pmatrix}\begin{pmatrix} -0.5 \\ -0.25 \end{pmatrix}=0$ 。

3.7 单纯形法

单纯形法又称可变多面体法。在 n 维空间 \mathbf{R}^n 中,单纯形是指具有 $n+1$ 个顶点的多面体。若多面体的各棱边的长度相等,则称为正规单纯形。例如,在二维空间 \mathbf{R}^2 中,三角形就是一个单纯形,而等边三角形是一个正规的单纯形。

单纯形法是对单纯形的 $n+1$ 个顶点的函数值进行比较,丢掉其中的最坏点[注:最坏点是指函数值最大的点,最好点是指函数值最小的点],代之以较好的新点,从而构成一个新的单纯形,使单纯形逐步逼近函数的极小点的一种搜索方法。

有几种初始单纯形的取法,下面,仅介绍常用的正规单纯形的取法。给定一个初始点 \mathbf{X}_0 ,按如下公式产生其余 n 点。单纯形的 $n+1$ 个顶点的具体公式为

$$\begin{aligned}
X_0 &= (a_1, a_2, \dots, a_n)^T \\
X_1 &= (a_1 + q, a_2 + p, \dots, a_n + p)^T \\
X_2 &= (a_1 + p, a_2 + q, \dots, a_n + p)^T \\
&\dots \quad \dots \quad \dots \\
X_n &= (a_1 + p, a_2 + p, \dots, a_n + q)^T
\end{aligned}$$

其中, $p = \frac{\sqrt{n+1}+1}{\sqrt{2n}}a$, $q = \frac{\sqrt{n+1}-n+1}{\sqrt{2n}}a$, a 为正规单纯形的棱长。

下面, 我们来讨论 p 和 q 的由来。

$$X_i - X_0 = (\overset{i}{p}, \dots, \overset{i}{p}, q, p, \dots, p)^T \quad (i=1, 2, \dots, n)$$

$$X_i - X_j = (0, \dots, 0, \overset{i}{q-p}, 0, \dots, 0, \overset{j}{p-q}, 0, \dots, 0)^T \quad (i \neq j, 1 \leq i, j \leq n)$$

由上面两式, 可得

$$\|X_i - X_0\|^2 = (n-1)p^2 + q^2 = a^2 \quad (3.7.1)$$

$$\|X_i - X_j\|^2 = 2(p-q)^2 = a^2 \quad (3.7.2)$$

不妨取 $p > q > 0$, 由式 (3.7.2), 得

$$p = q + \frac{a}{\sqrt{2}} \quad (3.7.3)$$

将上式代入式 (3.7.1), 求解可得

$$q = \frac{\sqrt{n+1}-n+1}{\sqrt{2n}}a \quad (3.7.4)$$

将上式代入式 (3.7.3), 求解可得

$$p = \frac{\sqrt{n+1}+1}{\sqrt{2n}}a \quad (3.7.5)$$

单纯形法的主要步骤如下:

- (1) 给定 X_0 、 a 和收敛精度 ε 。
- (2) 按正规单纯形产生其余 n 顶点 X_1, X_2, \dots, X_n 。
- (3) 计算单纯形各顶点的函数值 $f(X_i)$ ($i=0, 1, 2, \dots, n$)。
- (4) 求最坏点 X_H 、次坏点 X_G 和最好点 X_L

$$f(X_H) = \max \{f(X_i), \quad (0 \leq i \leq n)\}$$

$$f(X_G) = \max \{f(X_i), \quad (i \neq H, 0 \leq i \leq n)\}$$

$$f(X_L) = \min \{f(X_i), \quad (0 \leq i \leq n)\}$$

- (5) 求去掉最坏点以外其余点的形心 X_C , 即

$$X_C = \frac{1}{n} \sum_{i=0, i \neq H}^n X_i \quad (3.7.6)$$

(6) 求反射点 X_{n+1} , 即

$$X_{n+1} = X_C + \alpha(X_C - X_H) \quad (3.7.7)$$

其中 $\alpha(\alpha > 0)$ 为反射系数, 一般取 $\alpha = 1$ 。

二维情形反射过程的几何意义如图 3.7.1 所示。

(7) 延伸

若 $f(X_{n+1}) < f(X_L)$, 则沿 $X_{n+1} - X_C$ 方向延伸, 得到延伸点 X_{n+2} , 即

$$X_{n+2} = X_C + \gamma(X_{n+1} - X_C) \quad (3.7.8)$$

其中, $\gamma(\gamma > 1)$ 称为延伸系数, 一般取 $\gamma = 2$ 。若 $f(X_{n+2}) < f(X_L)$, 用 X_{n+2} 代替 X_H ; 否则, 用 X_{n+1} 代替 X_H , 转到步骤 (10)。

二维情形延伸过程的几何意义如图 3.7.2 所示。

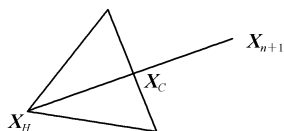


图 3.7.1 反射点

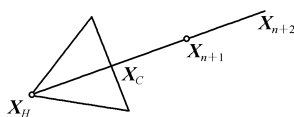


图 3.7.2 延伸点

(8) 若 $f(X_{n+1}) < f(X_G)$, 用 X_{n+1} 代替 X_H , 转到步骤 (11)。

(9) 收缩

① 若 $f(X_{n+1}) > f(X_H)$, 则令收缩点为

$$X_{n+3} = X_C + \beta(X_H - X_C) \quad (3.7.9)$$

② 若 $f(X_{n+1}) \leq f(X_H)$, 则令收缩点为

$$X_{n+3} = X_C + \beta(X_{n+1} - X_C) \quad (3.7.10)$$

其中, $\beta(0 < \beta < 1)$ 称为收缩系数, 常取 $\beta = 0.5$ 。

若 $f(X_{n+3}) < f(X_H)$, 用 X_{n+3} 代替 X_H , 转到步骤 (11)。

二维情形收缩过程的几何意义如图 3.7.3 所示。

(10) 缩边

若 $f(X_{n+3}) \geq f(X_H)$, 则将单纯形的各棱边的长度向最好点缩小一半, 即向量 $X_i - X_L$ 的长度缩小一半。有

$$X_i = X_L + \frac{(X_i - X_L)}{2} \quad (i=0,1,2,\dots,n) \quad (3.7.11)$$

二维情形缩边的几何意义如图 3.7.4 所示。

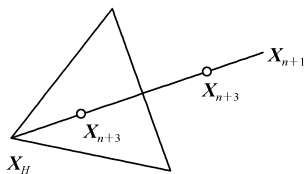


图 3.7.3 收缩

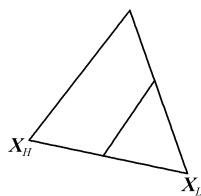


图 3.7.4 缩边

(11) 收敛准则

若 $\left\{ \frac{1}{n} \sum_{i=0}^n [f(X_i) - f(X_L)]^2 \right\}^{1/2} < \varepsilon$ 或 $f(X_H) - f(X_L) < \varepsilon$, 令 $X^* = X_L$, 打印最优解 X^* 和

$f(X^*)$, 停止程序执行; 否则转到步骤 (3)。

单纯形法程序框图如图 3.7.5 所示。

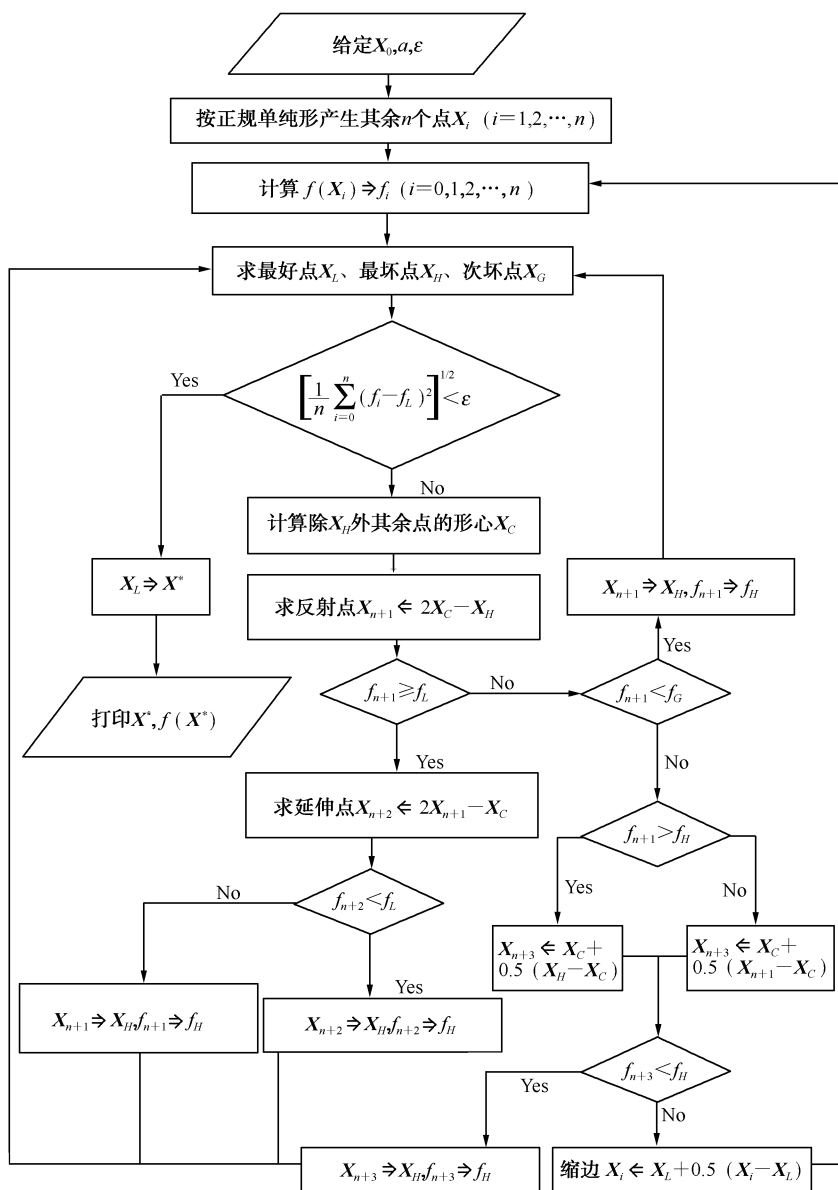


图 3.7.5 单纯形法程序框图

第 4 章 约束最优化问题的求解方法

本章讨论多维约束最优化问题，其一般形式为

$$\min f(\mathbf{X}) \quad (\mathbf{X} \in \mathbf{R}^n)$$

满足于

$$\begin{aligned} g_i(\mathbf{X}) &\leq 0 \quad (i=1,2,\cdots,m) \\ h_j(\mathbf{X}) &= 0 \quad (j=1,2,\cdots,l) \end{aligned} \quad (4.0.1)$$

式中， $g_i(\mathbf{X}) \leq 0$ ($i=1,2,\cdots,m$) 称为不等式约束条件， $h_j(\mathbf{X})=0$ ($j=1,2,\cdots,l$) 称为等式约束条件，不等式约束条件和等式约束条件统称为约束条件。

我们把满足所有约束条件的向量 $\mathbf{X} \in \mathbf{R}^n$ 称为可行点或可行解，所有可行点组成的集合称为可行解集，可行解集可表示为

$$\mathbf{D} = \{\mathbf{X}: g_i(\mathbf{X}) \leq 0 \quad (i=1,2,\cdots,m); h_j(\mathbf{X}) = 0 \quad (j=1,2,\cdots,l); \mathbf{X} \in \mathbf{R}^n\} \quad (4.0.2)$$

因此，约束优化问题可表示为

$$\min f(\mathbf{X}) \quad (\mathbf{X} \in \mathbf{D}) \quad (4.0.3)$$

求解约束优化问题 (4.0.1) 是指在可行解集 \mathbf{D} 中找一点 \mathbf{X}^* ，对于所有 $\mathbf{X} \in \mathbf{D}$ ，使

$$f(\mathbf{X}^*) \leq f(\mathbf{X}) \quad (4.0.4)$$

成立。点 \mathbf{X}^* 称为约束优化问题 (4.0.1) 的全局最优点。对于一般非线性约束最优化问题，求最优解是一件很复杂的工作。

求解约束优化问题的方法有^[9~13]：线性逼近法、罚函数法、半惩罚函数法、梯度投影法、可行方向法、网格法、复合形法、拉格朗日乘子法和增广拉格朗日乘子法等。下面介绍常用的几种约束优化问题的求解方法。

4.1 罚函数法

首先看一个例子，求解

$$\min \{x_1^2 + x_2^2\}$$

满足于

$$x_1 + x_2 - 2 = 0$$

用图解法或拉格朗日乘子法不难求出它的极小点 $\mathbf{X}^* = (1,1)^T$ 。

我们设想构造一个函数 $\varphi(\mathbf{X})$ ，当 $\mathbf{X} \in \mathbf{D}$ 时， $\varphi(\mathbf{X}) = f(\mathbf{X})$ ；当 $\mathbf{X} \notin \mathbf{D}$ 时， $\varphi(\mathbf{X})$ 为无穷大。对于上例，其构造函数 $\varphi(x_1, x_2)$ 为

$$\varphi(x_1, x_2) = \begin{cases} x_1^2 + x_2^2 & x_1 + x_2 - 2 = 0 \\ \infty & x_1 + x_2 - 2 \neq 0 \end{cases}$$

虽然, $\varphi(x_1, x_2)$ 的极小点一定落在直线 $x_1 + x_2 - 2 = 0$ 上。但由于 $\varphi(x_1, x_2)$ 的不连续性, 函数的特性太坏, 以致无法用无约束最优化方法求解。

考虑求如下函数的极小点:

$$\varphi(x_1, x_2, \gamma) = x_1^2 + x_2^2 + \gamma(x_1 + x_2 - 2)^2$$

式中, γ 为很大的正数。

$\varphi(x_1, x_2, \gamma)$ 可以用无约束最优化方法求解, 下面我们用解析法求解。令 $\mathbf{X}^*(\gamma)$ 为函数 $\varphi(x_1, x_2, \gamma)$ 的极小点, 则有

$$\nabla \varphi[\mathbf{X}^*(\gamma)] = 0$$

即

$$\begin{aligned} \frac{\partial \varphi}{\partial x_1} &= 2x_1 + 2\gamma(x_1 + x_2 - 2) = 0 \\ \frac{\partial \varphi}{\partial x_2} &= 2x_2 + 2\gamma(x_1 + x_2 - 2) = 0 \end{aligned}$$

解上式得

$$x_1^* = x_2^* = \frac{2\gamma}{2\gamma + 1}$$

取 $\gamma_1 = 10, \gamma_2 = 10\gamma_1, \dots, \gamma_{k+1} = 10\gamma_k$, 有如下序列:

$$x_1^*(\gamma_k) = \left\{ \frac{20}{21}, \frac{200}{201}, \frac{2000}{2001}, \frac{20000}{20001}, \frac{200000}{200001}, \dots \right\}$$

显然, 有

$$\lim_{k \rightarrow \infty} x_1^*(\gamma_k) = 1$$

上例说明约束最优化问题可以转化为无约束最优化问题来求解。将约束条件 $g_i(\mathbf{X})$ 和 $h_j(\mathbf{X})$ 以某种形式附加到目标函数上, 形成一个无约束最优化问题; 然后, 用无约束最优化方法求解, 这就是罚函数法。之所以称为罚函数法是因为对于 $\mathbf{X} \notin \mathbf{D}$ 的点, 对目标函数进行惩罚; 而对于 $\mathbf{X} \in \mathbf{D}$ 的点, 不对目标函数进行惩罚。根据惩罚函数的不同, 可分为外点罚函数法、内点罚函数法 (或称障碍函数法) 和扩展的内点罚函数法。

4.1.1 外点罚函数法

将约束最优化问题 (4.0.1) 转化为无约束最优化问题, 则

$$\min \varphi(\mathbf{X}, \gamma) \quad (4.1.1)$$

其中, $\varphi(\mathbf{X}, \gamma) = f(\mathbf{X}) + \gamma\alpha(\mathbf{X})$, γ 为任意大的正数。

函数 $\alpha(\mathbf{X})$ 应满足以下条件:

(1) $\alpha(\mathbf{X})$ 在 R^n 上的连续函数;

(2) 当 $X \notin D$ 时, $\alpha(X) > 0$;

(3) 当 $X \in D$ 时, $\alpha(X) = 0$ 。

一般称 $\alpha(X)$ 为惩罚项, γ 为惩罚因子, $\varphi(X, \gamma)$ 为惩罚函数。可取惩罚项 $\alpha(X)$ 为

$$\alpha(X) = \sum_{i=1}^m \{\max[0, g_i(X)]\}^2 + \sum_{j=1}^l h_j^2(X)$$

上述惩罚项 $\alpha(X)$ 的选取, 会使惩罚函数 $\varphi(X, \gamma)$ 的函数值、梯度在边界处连续, 但是, 二阶方向导数在边界处不连续。

在讨论算法之前, 我们首先研究约束最优化问题 (4.0.1) 与无约束最优化问题 (4.1.1) 的关系。

定理 4.1.1 若 X^* 是原优化问题 (4.0.1) 的极小点, 则 X^* 必为罚函数 $\varphi(X, \gamma)$ 的极小点。

证 因为 X^* 是问题 (4.0.1) 的解, 故 $X^* \in D$, 则有

$$\alpha(X^*) = 0$$

对于一切 $X \in D$, 有

$$\begin{aligned}\varphi(X^*, \gamma) &= f(X^*) + \gamma \alpha(X^*) \\ &\leq f(X) + \gamma \alpha(X) = \varphi(X, \gamma)\end{aligned}$$

对于一切 $X \notin D$, 有

$$\begin{aligned}\varphi(X^*, \gamma) &= f(X^*) + \gamma \alpha(X^*) \\ &\leq f(X^*) + \bar{\gamma} \alpha(X)\end{aligned}\quad (4.1.2)$$

式中, $\bar{\gamma} > 0$, 取一个的正数 γ 使 $\bar{\gamma} = \min \left\{ \gamma + \frac{f(X) - f(X^*)}{\alpha(X)} \right\} \geq 0$ 成立。有

$$\begin{aligned}\varphi(X^*, \gamma) &\leq f(X^*) + \bar{\gamma} \alpha(X) \\ &\leq f(X) + \gamma \alpha(X) \leq \varphi(X, \gamma)\end{aligned}$$

因此, X^* 是问题 (4.1.1) 的最优解, 证毕。

定理 4.1.2 对于任意固定的正数 γ_k , 设 $X^*(\gamma_k)$ 为罚函数 $\varphi(X, \gamma)$ 的最优解。如果 $X^*(\gamma_k) \in D$, 则 $X^*(\gamma_k)$ 必为原优化问题 (4.0.1) 的极小点。

证 对于一切 $X \in D$, 有 $\alpha(X) = 0$ 。由定理的条件, 有

$$\begin{aligned}f[X^*(\gamma_k)] &= \varphi[X^*(\gamma_k), \gamma_k] \\ &\leq \varphi[X, \gamma_k] = f(X)\end{aligned}$$

证毕。

定理 4.1.2 告诉我们, $\alpha[X(\gamma_k)] < \varepsilon$ 可作为罚函数算法的收敛准则, 其中 ε 为任意小的正数。

定理 4.1.3 假设 f, g_i ($i=1, 2, \dots, m$) 和 h_j ($j=1, 2, \dots, l$) 都是连续函数, 则由外点罚函数产生的任何收敛序列 $\{X_k\}$ 的极点必为原问题 (4.0.1) 的极小点。

定理的证明请参看文献[4]P324。

为了能用无约束最优化方法求解惩罚函数 $\varphi(X, \gamma)$, 惩罚函数 $\varphi(X, \gamma)$ 中的惩罚因子 γ 是逐步增大的。为了求得约束最优化问题的解, 需要求解一系列的无约束最优化问题的解, 因此, 罚函数法又称为序列无约束极小化方法, 即 SUMT (Sequential Unconstrained Minimization Technique) 法。

外点罚函数法的主要步骤如下:

(1) 给定初始点 \mathbf{X}_0 , \mathbf{X}_0 可以不属于可行解集 \mathbf{D} , 选取初始惩罚因子 $\gamma_1 > 0$, 一般取 $\gamma_1 = 10$, 惩罚因子放大系数 $c > 0$, 一般取 $c = 10$, 置 $k = 1$ 。

(2) 以 \mathbf{X}_{k-1} 为初始点, 求解如下无约束最优化问题:

$$\min \{f(\mathbf{X}) + \gamma_k \alpha(\mathbf{X})\}$$

其中, $\alpha(\mathbf{X}) = \sum_{i=1}^m \{\max[0, g_i(\mathbf{X})]\}^2 + \sum_{j=1}^l h_j^2(\mathbf{X})$, 设其极小点为 \mathbf{X}_k 。

(3) 若 $\alpha(\mathbf{X}_k) < \varepsilon$, 则 \mathbf{X}_k 就是所求的最优点 \mathbf{X}^* , 打印最优解 \mathbf{X}^* 和 $f(\mathbf{X}^*)$, 停止程序执行; 否则转到步骤 (4)。

(4) 置 $\gamma_{k+1} = c\gamma_k$, $k = k + 1$, 转到步骤 (2)。

可以用任意一种求解无约束最优化问题的方法求解 $\min \{f(\mathbf{X}) + \gamma_k \alpha(\mathbf{X})\}$, 但选用的方法不同, 直接影响算法的收敛速度, 当然, 我们应当选择收敛速度快的无约束最优化方法。

罚函数法程序框图如图 4.1.1 所示。

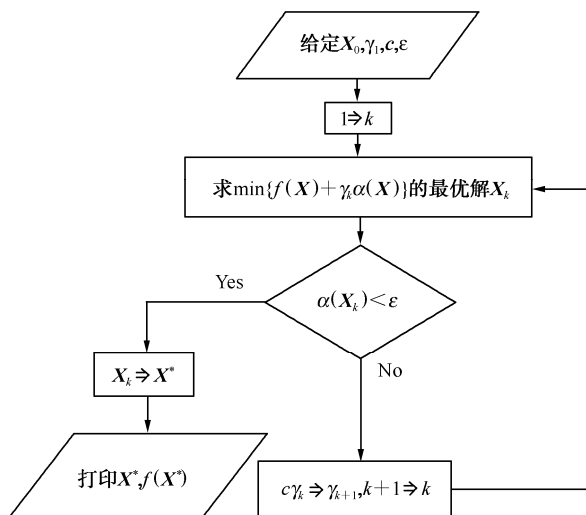


图 4.1.1 罚函数法程序框图

例 4.1.1 用外点罚函数法求解如下二维优化问题:

$$\min f(x) = \frac{(x+2)^2}{20}$$

满足于

$$g_1(x) = \frac{(1-x)}{2} \leq 0$$

$$g_2(x) = \frac{(x-2)}{2} \leq 0$$

解

$$\varphi(x, \gamma) = \frac{(x+2)^2}{20} + \gamma \left\{ \left[\max \left(0, \frac{1-x}{2} \right) \right]^2 + \left[\max \left(0, \frac{x-2}{2} \right) \right]^2 \right\}$$

上式等价于

$$\varphi(x, \gamma) = \begin{cases} \frac{(x+2)^2}{20} + \gamma \frac{(1-x)^2}{4} & (x < 1) \\ \frac{(x+2)^2}{20} & (1 \leq x \leq 2) \\ \frac{(x+2)^2}{20} + \gamma \frac{(x-2)^2}{4} & (x > 2) \end{cases}$$

分别令

$$f_1(x) = \frac{(x+2)^2}{20} + \gamma \frac{(1-x)^2}{4} \quad (x < 1)$$

$$f_2(x) = \frac{(x+2)^2}{20} \quad (1 \leq x \leq 2)$$

$$f_3(x) = \frac{(x+2)^2}{20} + \gamma \frac{(x-2)^2}{4} \quad (x > 2)$$

分别求函数 $f_1(x)$ 、 $f_2(x)$ 和 $f_3(x)$ 的极小点，它们的极小点分别为

$$x_1^*(\gamma) = \frac{5\gamma - 2}{5\gamma + 1}$$

$$x_2^*(\gamma) = 1$$

$$x_3^*(\gamma) = 2$$

式中， $x_i^*(\gamma)$ 为函数 $f_i(x)$ ($i=1,2,3$) 的极小点。

比较 $f_1(x)$ 、 $f_2(x)$ 和 $f_3(x)$ 极小点的函数值，得到函数 $\varphi(x, \gamma)$ 的极小点为

$$x(\gamma) = \frac{5\gamma - 2}{5\gamma + 1}$$

函数 $\varphi(x, \gamma)$ 在极小点处的函数值为

$$\varphi[x^*(\gamma)] = \frac{9}{20} - \frac{36\gamma}{20(5\gamma + 1)^2}$$

取 $\gamma = \{1, 10, 100, 1000, \dots\}$ ，有

$$x^*(\gamma) = \left\{ \frac{1}{2}, \frac{48}{51}, \frac{498}{501}, \frac{4998}{5001}, \dots \right\}$$

$$\lim_{k \rightarrow \infty} x^*(\gamma_k) = 1$$

$$\lim_{k \rightarrow \infty} \varphi[x^*(\gamma_k)] = \frac{9}{20}$$

图 4.1.2 和图 4.1.3 分别表示了例 4.1.1 优化问题的目标函数、约束条件和惩罚函数等情况。

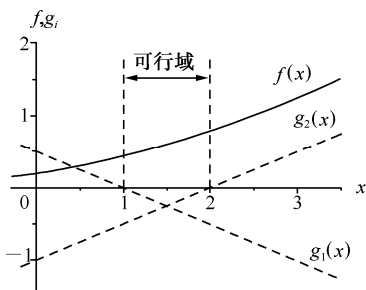
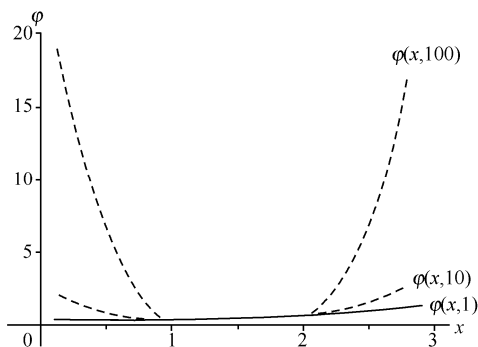


图 4.1.2 目标函数、约束条件和可行域

图 4.1.3 外点罚函数法中 γ 与 $\varphi(X, \gamma)$ 的关系曲线

用外点罚函数法求解时,初始点 \mathbf{X}_0 可以不属于可行域 \mathbf{D} , 这给算法带来了很大的便利; 但另一方面, 从例 4.1.1 可以看出, 当惩罚因子增大时, $\mathbf{X}^*(\gamma) \rightarrow \mathbf{X}^*$, 但 $\mathbf{X}^*(\gamma) \notin \mathbf{D}$, 只有当 $\gamma \rightarrow \infty$ 时, $\mathbf{X}^*(\gamma) \in \mathbf{D}$ 。但实际计算不可能取 γ 为无穷大, 用外点罚函数法求得的 $\mathbf{X}^*(\gamma)$ 有可能为不切实际的解。这一点被认为是外点罚函数法的缺点, 而内点罚函数法能克服这一缺点。

4.1.2 内点罚函数法

内点罚函数法与外点罚函数法不同之处有两点, 其一是对不等式约束函数取的惩罚函数的形式不同, 其二是惩罚因子在优化过程中是下降的。内点罚函数法的初始点必须满足所有不等式约束条件。内点罚函数法的基本思想是对不等式约束, 在接近可行解集的边界的点施加越来越大的惩罚, 对可行解集的边界的点施加无穷大的惩罚。

对不等式约束, 内点罚函数的惩罚项为

$$\alpha_1(\mathbf{X}) = \sum_{i=1}^m \frac{-1}{g_i(\mathbf{X})} \quad (4.1.3)$$

利用式 (4.1.3), 并且包含外点罚函数法对等式约束的惩罚项, 则内点罚函数法的惩罚函数为

$$\varphi(\mathbf{X}, \bar{\gamma}, \gamma) = f(\mathbf{X}) + \bar{\gamma} \sum_{i=1}^m \frac{-1}{g_i(\mathbf{X})} + \gamma \sum_{j=1}^l h_j^2(\mathbf{X}) \quad (4.1.4)$$

其中, 惩罚因子 $\bar{\gamma}$ ($\bar{\gamma} > 0$) 在优化过程中是下降的, 而 γ ($\gamma > 0$) 在优化过程中是增大的。

式 (4.1.3) 的另一种形式为

$$\alpha_1(\mathbf{X}) = \sum_{i=1}^m -\log[-g_i(\mathbf{X})] \quad (4.1.5)$$

由于式 (4.1.5) 数值条件数比式 (4.1.3) 稍好, 因此, 经常被推荐使用。

内点罚函数法的主要步骤如下:

- (1) 给定初始点 \mathbf{X}_0 、初始惩罚因子 $\bar{\gamma}_1$ 、 γ_1 和惩罚因子系数 c ($0 < c < 1$), 置 $k=1$ 。
- (2) 以 \mathbf{X}_{k-1} 为初始点, 求解如下无约束最优化问题:

$$\min \left\{ f(\mathbf{X}) + \bar{\gamma}_k \sum_{i=1}^m \frac{-1}{g_i(\mathbf{X})} + \gamma_k \sum_{j=1}^l h_j^2(\mathbf{X}) \right\}$$

设其极小点为 \mathbf{X}_k 。

(3) 若 $\|\mathbf{X}_k - \mathbf{X}_{k-1}\| < \varepsilon_1$ 和 $f(\mathbf{X}_{k-1}) - f(\mathbf{X}_k) < \varepsilon_2$, 则 \mathbf{X}_k 就是所求的最优点 \mathbf{X}^* , 打印最优解 \mathbf{X}^* 和 $f(\mathbf{X}^*)$, 停止程序执行; 否则转到步骤 (4)。

(4) 置 $\bar{\gamma}_{k+1} = c\bar{\gamma}_k$, $\gamma_{k+1} = \gamma_k/c$, $k = k+1$, 转到步骤 (2)。

内点罚函数法, 要求初始点 \mathbf{X}_0 为可行点, 即 $\mathbf{X}_0 \in \mathbf{D}$ 。下面介绍求解约束最优化问题可行点的两种方法。

第一种求解约束最优化问题可行点的方法的主要步骤为^[3]:

(1) 给定初始点 \mathbf{X}_0 , 置 $k=0$ 。

(2) 设 $I = \{i: g_i(\mathbf{X}_k) < 0\}$, 如果 $I = \{1, 2, \dots, m\}$, 则 \mathbf{X}_k 为所求的可行点; 否则, 选一个 $j \notin I$, 转到步骤 (3)。

(3) 用内点罚函数法求解如下优化问题:

$$\min \{g_j(\mathbf{X})\} \quad (4.1.6)$$

满足于

$$g_i(\mathbf{X}) < 0 \quad (i \in I)$$

设 \mathbf{X}_{k+1} 为上述优化问题的解, 如果 $g_j(\mathbf{X}_{k+1}) \geq 0$, 则可行解集为空集, 停止程序执行; 否则, 置 $k = k+1$, 转到步骤 (2)。

第二种求解约束最优化问题可行点的方法是作者在文献[14]中提出的一种求解最优化问题可行解的快速有效方法, 现将该方法叙述如下。

因为内点罚函数法要求初始点 $\bar{\mathbf{X}}_0$ 满足 $g_i(\bar{\mathbf{X}}_0) < 0 \quad (i=1, 2, \dots, m)$, 将不等式约束 $g_i(\mathbf{X}) \leq 0$ 转化为 $\bar{g}_i(\mathbf{X}) = g_i(\mathbf{X}) + \varepsilon < 0 \quad (i=1, 2, \dots, m)$ 。其中 ε 为较小的正数。 ε 的值由具体问题确定, 一般可取 $\varepsilon = 0.01 \sim 0.1$ 。

为了求得满足所有不等式约束的可行点, 建立如下函数:

$$\alpha_2(\mathbf{X}) = \sum_{i=1}^m \left\{ \max[0, \bar{g}_i(\mathbf{X})] \right\}^2 \quad (4.1.7)$$

$\alpha_2(\mathbf{X})$ 具有函数和一阶导数连续, 以及满足约束条件 $\bar{g}_i(\mathbf{X}) = g_i(\mathbf{X}) + \varepsilon < 0 \quad (i=1, 2, \dots, m)$ 的所有 $\bar{\mathbf{X}}$ 都是函数 $\alpha_2(\mathbf{X})$ 的全局最小点的特点, 且有 $\alpha_2(\bar{\mathbf{X}}) = 0$ 。

第二种求解约束最优化问题可行点的方法的主要步骤:

(1) 初始点 \mathbf{X}_0 。

(2) 以 \mathbf{X}_0 为初始点, 求解

$$\min \alpha_2(\mathbf{X}) = \min \sum_{i=1}^m \left\{ \max[0, \bar{g}_i(\mathbf{X})] \right\}^2$$

设 \mathbf{X}_1 为上述优化问题的解, 如果 $\alpha_2(\mathbf{X}_1) > 0$, 则可行解集为空集; 若 $\alpha_2(\mathbf{X}_1) = 0$, 则 \mathbf{X}_1 为一个可行点。

与第一种求解约束最优化问题可行点的方法比较, 第二种求解约束最优化问题可行点的方法有两个优点。第一个优点是整个约束优化问题的可行解集都是优化函数 $\alpha_2(\mathbf{X})$ 的最优解, 而方法一的 $g_j(\mathbf{X})$ 函数最优解一般只有一个最优点, 因此, 与方法一比较方法二更容易求解。第二个优点是计算时间

大幅度减少, 设方法一需要求解最优化问题 (4.1.6) 为 k_1 次, 平均每次用内点罚函数法求解最优化问题 (4.1.6) 需要 k_2 次无约束最优化问题的求解, 用方法一共需要 $k_1 \times k_2$ 次无约束最优化问题的求解; 而方法二只需要一次无约束最优化问题的求解。同时, 考虑到方法一的最优点一般只有一个点, 而方法二的最优点是整个可行解集, 因此, 方法二的计算量不到方法一的计算量的 $k_1 \times k_2$ 分之一。

作者用 Powell 法作为无约束最优化问题的求解方法, 编写了求解约束最优化问题可行解方法二的程序, 用程序计算的数例如下。

例 4.1.2 求第 1 章利润问题的可行解, 即

$$g_1(X) = 3x_1 + 4x_2 - 12 \leq 0$$

$$g_2(X) = 3x_1 + 3x_2 - 10 \leq 0$$

$$g_3(X) = 4x_1 + 2x_2 - 8 \leq 0$$

$$g_4(X) = -x_1 \leq 0$$

$$g_5(X) = -x_2 \leq 0$$

取初始点 $X_0 = (90, -90)^T$, 用程序求得的可行点为 $\bar{X} = (0.6172, 0.556)^T$ 。

例 4.1.3 用内点罚函数法求例 4.1.1。

解

$$\varphi(x, \bar{\gamma}) = f(X) + \bar{\gamma} \sum_{i=1}^m \frac{-1}{g_i(X)} = \frac{(x+2)^2}{20} - \bar{\gamma} \left(\frac{2}{1-x} + \frac{2}{x-2} \right)$$

$$\varphi'(x, \bar{\gamma}) = \frac{(x+2)}{10} - \bar{\gamma} \left(\frac{2}{(1-x)^2} - \frac{2}{(x-2)^2} \right) = 0$$

取 $\bar{\gamma} = \{0.5, 0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001\}$, 分别将 $\bar{\gamma}$ 代入上式, 得

$$x^*(\bar{\gamma}) \approx \{1.4895, 1.4475, 1.2365, 1.0805, 1.0255, 1.0085, 1.0025\}$$

对于不同的惩罚因子 $\bar{\gamma}$, $\varphi(X, \bar{\gamma})$ 的函数值如图 4.1.4 所示。

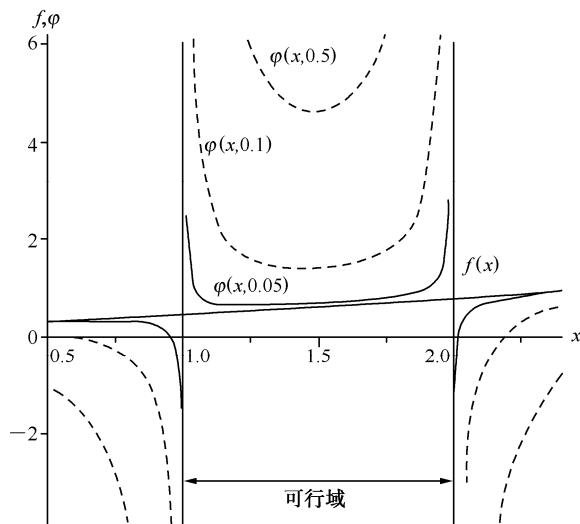


图 4.1.4 内点罚函数法中 $\bar{\gamma}$ 与 $\varphi(X, \bar{\gamma})$ 的关系曲线

内点罚函数法除了要求初始点 $X_0 \in D$ 以外, 内点罚函数法对无约束最优化算法要求很高。从图 4.1.4 可以看出, 一不小心, 如果迭代点进入非可行域, 就会导致无法求得最优解, 从非

可行域靠近边界时, $\varphi(\mathbf{X}, \bar{\gamma})$ 的函数值为负无穷大。

4.1.3 扩展的内点罚函数法

扩展的内点罚函数法充分利用了内、外点罚函数法的优点, 克服了它们的缺点。这种方法处理等式约束与外点罚函数法相同, 因此, 为了简单, 在下面的讨论中, 省略等式约束的处理。

1. 线性扩展罚函数

我们取惩罚项为

$$\alpha_1(\mathbf{X}) = \sum_{i=1}^m \bar{g}_i(\mathbf{X}) \quad (4.1.8)$$

$$\text{式中, } \bar{g}_i(\mathbf{X}) = \begin{cases} -1 & g_i(\mathbf{X}) \leq \varepsilon \\ g_i(\mathbf{X}) & \\ -\frac{2\varepsilon - g_i(\mathbf{X})}{\varepsilon^2} & g_i(\mathbf{X}) > \varepsilon \end{cases} \quad (4.1.9)$$

ε 为一个很小的负数, 文献[11]推荐按下式计算:

$$\varepsilon = -c\gamma_k^\alpha \quad \left(\frac{1}{3} \leq \alpha \leq \frac{1}{2}\right)$$

式(4.1.8)定义的函数 $\alpha_1(\mathbf{X})$ 在 $g_i(\mathbf{X}) = \varepsilon$ 处连续, 且一阶导数也连续; 但二阶导数不连续。

例 4.1.4 用线性扩展的内点罚函数法求解例 4.1.1。

解 取 $\bar{\gamma} = \{0.5, 0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001\}$, $c = 0.15$, $a = 0.5$, 用线性扩展的内点罚函数法求解, 得

$$x^*(\bar{\gamma}) \approx \{1.489, 1.447, 1.236, 1.080, 1.026, 1.008, 1.003\}$$

对于 $c = 0.15$ 和 $a = 0.5$, 线性扩展罚函数的情形如图 4.1.5 所示。

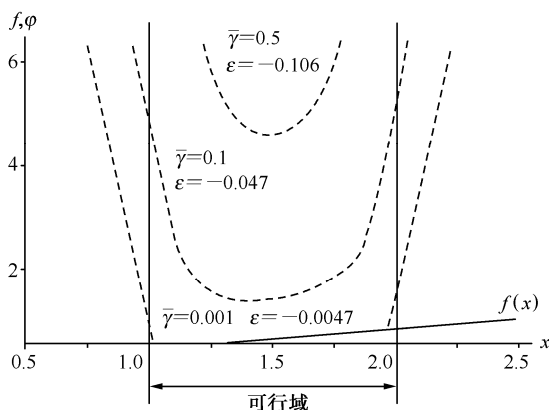
从例 4.1.3 和例 4.1.4 可知, 用内点罚函数法和用线性扩展的内点罚函数法求得的最优解 $x^*(\bar{\gamma})$ 几乎相同, 但线性扩展的内点罚函数法所用的罚函数的特性较好, 函数和一阶导数连续, 这大大降低了对无约束最优化算法的要求, 初始点 \mathbf{X}_0 也可以选择不属于可行域的点。

2. 二次扩展罚

在式(4.1.8)中, 取 $\bar{g}_i(\mathbf{X})$ 为

$$\bar{g}_i(\mathbf{X}) = \begin{cases} -1 & g_i(\mathbf{X}) \leq \varepsilon \\ g_i(\mathbf{X}) & \\ -\frac{1}{\varepsilon} \left[\left(\frac{g_i(\mathbf{X})}{\varepsilon} \right)^2 - 3 \left(\frac{g_i(\mathbf{X})}{\varepsilon} \right) + 3 \right] & g_i(\mathbf{X}) > \varepsilon \end{cases} \quad (4.1.10)$$

式(4.1.10)定义的函数 $\alpha_1(\mathbf{X})$ 在 $g_i(\mathbf{X}) = \varepsilon$ 处具有函数、一阶导数和二阶导数连续性; 但 $\varphi(\mathbf{X}, \bar{\gamma}, \gamma)$ 函数的非线性增大。因此, 用一阶导数的无约束最优化方法求解 $\varphi(\mathbf{X}, \bar{\gamma}, \gamma)$ 时, 建议采用线性扩展函数。

图 4.1.5 线性扩展内点罚函数法中 $\bar{\gamma}$ 与 $\varphi(X, \bar{\gamma})$ 的关系曲线

4.2 复合形法

复合形法是求解约束最优化问题的直接方法之一，是无约束最优化问题的单纯形法的推广。它只适用于只有不等式约束的最优化问题，即

$$\min f(X) \quad (4.2.1)$$

满足于

$$g_i(X) \leq 0 \quad (i=1, 2, \dots, m)$$

用单纯形法求解约束最优化问题，在处理约束条件时会出现一些困难。当向边界调优时，单纯形法容易退化降维，当反射点 $X_R \approx X_C$ 时，产生降维现象，如图 4.2.1 所示。一旦产生了降维现象，就不能求得最优化问题的最优解。复合形法是以 n_1 个初始点为顶点构成一个不规则的多面体，为了防止退化降维现象，复合形法的多面体的顶点数 n_1 要大于 $n+1$ ，一般 $n+1 < n_1 \leq 2n$ 。

复合形法的基本思想是先在可行域内产生一个具有 n_1 个顶点的初始复合形（对于低维优化问题， n_1 可以多于 $2n$ 个点；对于高维优化问题， n_1 可以少于 $2n$ 个点，但 n_1 必须多于 $n+1$ 个点）。对复合形顶点的函数进行比较，不断地丢掉其中的最坏点，代之以较好的新点，构成新的复合形，从而逐步使复合形逼近函数的极小点。复合形法的主要步骤如下：

(1) 初始复合形的 n_1 个顶点的产生。产生 n_1 个顶点的方法有许多，下面，我们介绍输入一个初始点，其余 n_1-1 个点用随机函数产生的方法。

① 输入一个可行的初始点 X_1 。

② 产生其余的 n_1-1 个点。

用下式产生 X_{k+1} ($k=1, 2, \dots, n_1-1$) 点：

$$x_{k+1,i} = a_i + \zeta(b_i - a_i) \quad (4.2.2)$$

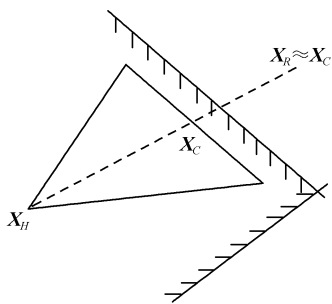


图 4.2.1 产生降维示意图

式中, $x_{k+1,i}$ 是 \mathbf{X}_{k+1} 第 i 个分量, a_i 和 b_i 分别为 x_i 的最优解估计的下限和上限, ζ 为 $0 \sim 1$ 之间的随机数。

如果 \mathbf{X}_{k+1} 满足约束条件, 继续产生下一个顶点; 否则, 求前 k 个点的形心

$$\mathbf{X}_C = \frac{1}{k} \sum_{i=1}^k \mathbf{X}_i \quad (4.2.3)$$

将 \mathbf{X}_{k+1} 沿 \mathbf{X}_{k+1} 与 \mathbf{X}_C 的连线向形心点 \mathbf{X}_C 缩小一半, 即

$$\mathbf{X}_{k+1} = \mathbf{X}_C + 0.5(\mathbf{X}_{k+1} - \mathbf{X}_C) \quad (4.2.4)$$

然后, 再验证 \mathbf{X}_{k+1} 是否满足约束条件。如果仍然不满足, 再按上式计算, 再验证 \mathbf{X}_{k+1} 是否满足约束条件, 如此重复, 直到 \mathbf{X}_{k+1} 满足约束条件为止。

如果 \mathbf{X}_C 不满足约束条件, 即可行域为非凸集, 则上面的方法是无效的, 使程序陷入死循环。作者建议当可行域为非凸集时, 采用罚函数法求解, 或者重新产生 \mathbf{X}_{k+1} 点。

(2) 计算复合形各顶点的函数值

$$f_i = f(\mathbf{X}_i) \quad (i=1, 2, \dots, n_1) \quad (4.2.5)$$

(3) 求最坏点 \mathbf{X}_H 和最好点 \mathbf{X}_L

$$f(\mathbf{X}_H) = \max \{f_i, (1 \leq i \leq n_1)\}$$

$$f(\mathbf{X}_L) = \min \{f_i, (1 \leq i \leq n_1)\}$$

(4) 若 $\left[\frac{1}{n_1 - 1} \sum_{i=1}^{n_1} (f_i - f_L)^2 \right]^{1/2} < \varepsilon$ (ε 为控制收敛精度的最小的正数), 令 $\mathbf{X}^* = \mathbf{X}_L$, 打印最

优解 \mathbf{X}^* 和 $f(\mathbf{X}^*)$, 停止程序执行; 否则, 转到步骤 (5)。

(5) 求去掉最坏点以外其余点的形心

$$\mathbf{X}_C = \frac{1}{n_1 - 1} \sum_{i=1, i \neq H}^{n_1} \mathbf{X}_i \quad (4.2.6)$$

如果 \mathbf{X}_C 不满足约束条件, 令 $\mathbf{A} = \mathbf{X}_L$, $\mathbf{B} = \mathbf{X}_H$ 和 $\mathbf{X}_1 = \mathbf{X}_L$, 转到步骤 (1), 重新产生新的复合形。

(6) 求反射点

$$\mathbf{X}_R = \mathbf{X}_C + \alpha(\mathbf{X}_C - \mathbf{X}_H) \quad (4.2.7)$$

其中, $\alpha (\alpha > 0)$ 为反射系数, 一般取 $\alpha = 1.3$ 。

(7) 验证 \mathbf{X}_R 是否是可行点。若 \mathbf{X}_R 不是可行点, 则置 $\alpha = \frac{\alpha}{2}$, 转到步骤 (6)。

(8) 计算 $f(\mathbf{X}_R)$, 如果 $f(\mathbf{X}_R) < f(\mathbf{X}_H)$, 则用反射点 \mathbf{X}_R 代替最坏点 \mathbf{X}_H , 构成新的复合形, 转到步骤 (3); 如果 $f(\mathbf{X}_R) \geq f(\mathbf{X}_H)$, 转到步骤 (9)。

(9) 置 $\alpha = \frac{\alpha}{2}$, 若 $\alpha \leq \delta$ (δ 为预先给定的一个小的正数), 转到步骤 (10); 否则转到步骤 (6)。

(10) 如果上次不是用次坏点代替最坏点, 则用次坏点 \mathbf{X}_G

$$f(\mathbf{X}_G) = \max \{f(\mathbf{X}_i) | (1 \leq i \leq n_1, i \neq H)\} \quad (4.2.8)$$

代替最坏点 \mathbf{X}_H , 转到步骤 (3); 否则, 转到步骤 (11)。

(11) 缩边。

复合形各顶点向最好点 X_L 收缩二分之一，即

$$X_i = X_L + 0.5(X_i - X_L) \quad (i=1, 2, \dots, n_1) \quad (4.2.9)$$

验证 X_i 是否为可行点，若某个 X_i 不是可行点，建议按复合形法的第 (5) 步处理；若所有 X_i 是可行点，转到步骤 (2)。

复合形法程序框图如图 4.2.2 所示。

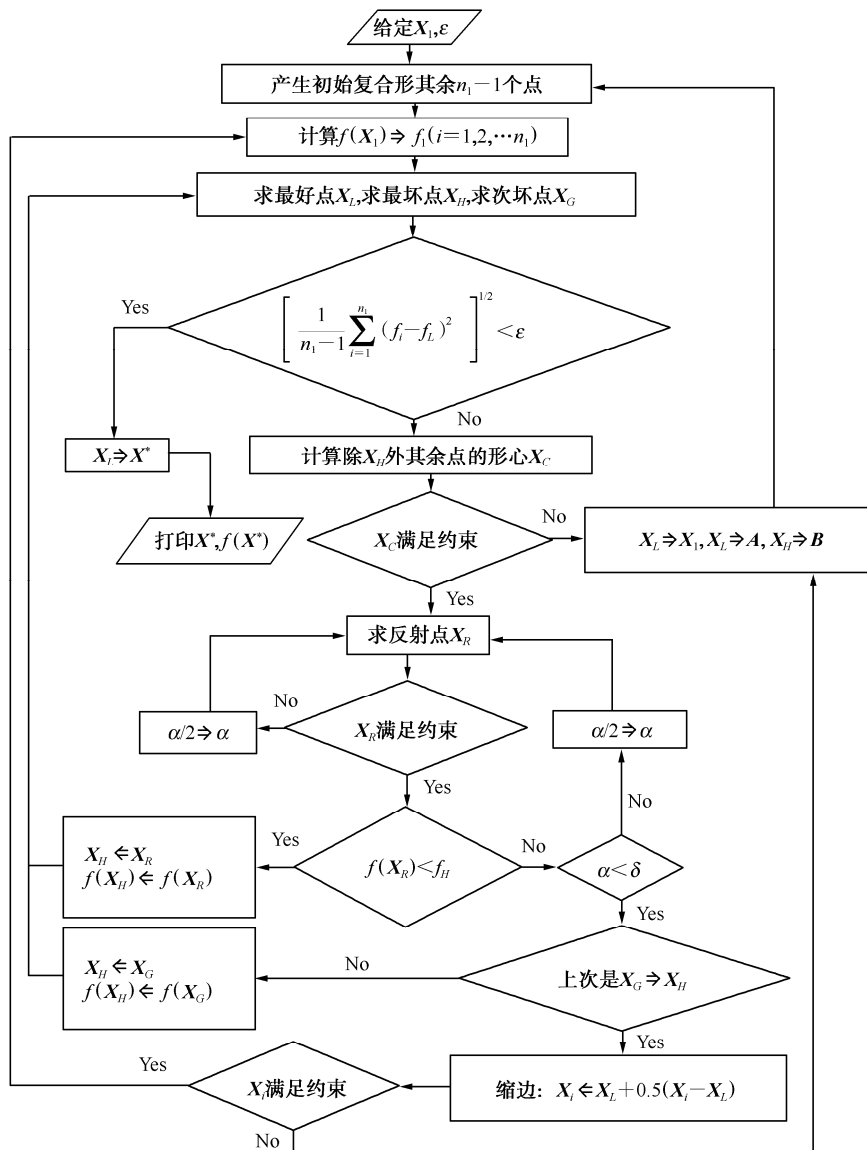


图 4.2.2 复合形法程序框图

例 4.2.1 用复合形法求第 1 章利润问题的最优解。

$$\max \{4x_1 + 3x_2\}$$

满足于

$$3x_1 + 4x_2 \leq 12$$

$$3x_1 + 3x_2 \leq 10$$

$$4x_1 + 2x_2 \leq 8$$

$$x_1, x_2 \geq 0$$

用作者编写的复合形法程序计算的最优解为 $\mathbf{X}^* = (0.8000025, 2.399994)^T$, $f(\mathbf{X}^*) = 10.39999$
[注: 精确解为 $\mathbf{X}^* = (0.8, 2.4)^T$, $f(\mathbf{X}^*) = 10.4$].

4.3 半惩罚函数法^[15]

4.3.1 半惩罚函数法提出的原因

作者在研究双动拉延压力机内滑块十杆机构的优化设计(该机构的优化设计将在第 8 章 8.7 节中讨论)时发现: ① 很难求得该机构的最优化设计问题的可行点; ② 即使求到了初始可行点或输入一个可行的初始点, 后面的优化设计过程也无法进行下去。分析其原因有两个, 其一是因为该机构复杂, 约束条件较多, 并且, 它们之间相互联系; 另一个更重要的原因是当破坏约束条件时, 会使其他约束条件或计算目标函数中的 $\arccos x$ 中 x 的绝对值大于 1, 致使计算程序中断。

当 $\arccos x$ 中 x 的绝对值大于 1 时, 用罚函数法计算难以处理; 用复合形法计算, 又找不到可行点, $\arccos x$ 中 x 的绝对值大于 1 也难以处理; 其他最优化方法处理这样的最优化问题也存在困难。

4.3.2 半惩罚函数法

首先, 将约束最优化问题中的不等式约束条件分为两类。第一类约束条件为破坏约束条件会引起计算程序中断的约束条件, 例如, 如果拉延压力机内滑块十杆机构优化设计中某些

约束条件不满足, 十杆机构优化设计中的约束条件 $\phi_{\min} - \arccos \frac{l_2^2 + l_3^2 - (C - l_1)^2}{2l_2l_3} \leq 0$ (见第 8

章 8.7 节) 中 $\left| \frac{l_2^2 + l_3^2 - (C - l_1)^2}{2l_2l_3} \right| > 1$, 导致计算程序中断。用 I_1 来表示所有第一类约束条件

的下标的集合, 即 $I_1 = \{i: g_i(\mathbf{X}) \leq 0, g_i(\mathbf{X}) \leq 0 \text{ 为第一类约束条件}\}$ 。第二类约束条件为破坏约束条件不会引起计算程序中断的约束条件。用 I_2 来表示所有第二类约束条件的下标的集合, 即 $I_2 = \{i: g_i(\mathbf{X}) \leq 0, g_i(\mathbf{X}) \leq 0 \text{ 为第二类约束条件}\}$ 。

其次, 将第二类不等式约束条件和等式约束条件按照外点罚函数法来处理, 将其惩罚项加到目标函数中形成惩罚函数, 而保留第一类约束条件, 形成新的最优化问题, 即

$$\min \varphi(\mathbf{X}, \gamma_k) = f(\mathbf{X}) + \gamma_k \alpha(\mathbf{X}) \quad (4.3.1)$$

满足于

$$g_i(\mathbf{X}) \leq 0 \quad (i \in I_1)$$

式中, γ_k 为外点罚函数法中的惩罚因子, 惩罚项 $\alpha(\mathbf{X})$ 为

$$\alpha(\mathbf{X}) = \sum_{i \in I_2} \{\max[0, g_i(\mathbf{X})]\}^2 + \sum_{j=1}^l h_j^2(\mathbf{X})$$

最后, 我们用复合形法求解新的最优化问题 (4.3.1)。因为, 约束条件中的一部分约束条件用惩罚函数的方法转化为目标函数, 而另一部分约束条件继续保留下来, 因此, 作者将这种优化方法称为半惩罚函数法。

半惩罚函数法之所以能够求解双动拉延压力机内滑块十杆机构这类最优化问题, 是因为将一部分约束条件用惩罚项转化到目标函数中, 这样使得求解新的最优化问题的可行点变得容易; 同时, 保留了破坏约束条件会引起计算程序中断的第一类约束条件。

半惩罚函数法的主要步骤:

(1) 给定满足第一类约束条件的任意初始点 \mathbf{X}_0 、初始惩罚因子 γ_0 、 c 、 ε 和 n_{\max} (n_{\max} 为在惩罚因子 γ_k 下, 计算函数最好点的最大次数, 即将 n_{\max} 作为一个收敛准则), 令 $k=0$ 。

(2) 以 \mathbf{X}_k 作为复合形的一个顶点, 用复合形法求解最优化问题 (4.3.1), 设 \mathbf{X}_{k+1} 为所求的解。

(3) 若 $\alpha(\mathbf{X}_{k+1}) < \varepsilon$, 令 $\mathbf{X}^* = \mathbf{X}_{k+1}$, 打印最优解 \mathbf{X}^* 和 $f(\mathbf{X}^*)$, 停止程序执行; 否则, 转到步骤 (4)。

(4) 置 $\gamma_{k+1} = c\gamma_k$, $k = k+1$, 转到步骤 (2)。

半惩罚函数法程序框图如图 4.3.1 所示。

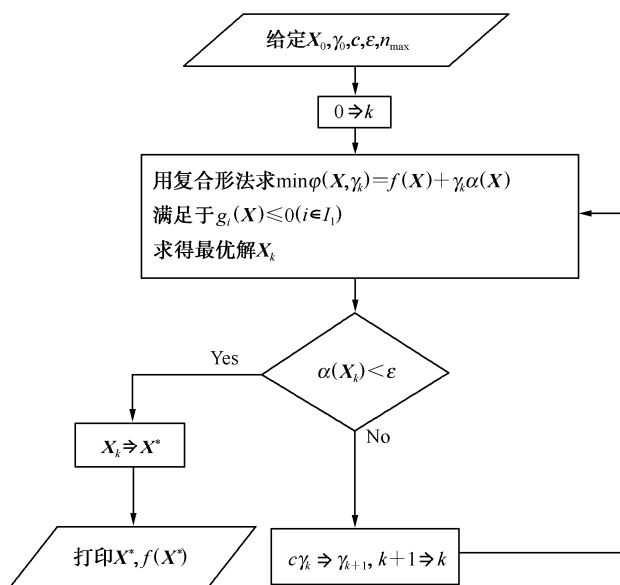


图 4.3.1 半惩罚函数法程序框图

4.4 增广拉格朗日乘子法^[16]

在本章的4.1节中,我们讨论了三种罚函数法求解约束最优化问题的古典方法,这些方法容易编程,且被认为是非常有效的方法。我们注意到引入 Kuhn-Tucker 条件方程式(1.8.6)中的拉格朗日乘子到罚函数法中,这些方法会得到很大的改进,著名数学家 Powell 指出:没有包含拉格朗日乘子的 SUMT 算法作为一种实用方法已经过时了。

在这一节中,我们将讨论如何将拉格朗日乘子引入到算法中来改进算法的有效性和可靠性。主要目的是减少算法对惩罚因子的依赖性。下面,我们讨论增广拉格朗日乘子法(ALM法)。ALM法首先用于解等式约束的最优化问题,然后,扩展到不等式约束的最优化问题,最后是两种方法合并起来求解一般约束最优化问题。

4.4.1 等式约束最优化问题

ALM法首先用于解等式约束的最优化问题

$$\min f(\mathbf{X}) \quad (4.4.1)$$

$$\text{满足于} \quad h_j(\mathbf{X})=0 \quad (j=1,2,\dots,l) \quad (4.4.2)$$

很显然,为了使上述最优化问题有意义, l 必须小于 n (n 为最优化问题的维数),即 $l < n$;当 $l=n$ 时,由式(4.4.2)就能确定唯一的可行点 \mathbf{X}^* ,当然, \mathbf{X}^* 就是最优点;当 $l > n$ 时,该最优化问题变得无意义。

Kuhn-Tucker 条件方程式(1.8.6)等价于建立如下拉格朗日函数:

$$L(\mathbf{X}, \boldsymbol{\lambda}) = f(\mathbf{X}) + \sum_{j=1}^l \lambda_j h_j(\mathbf{X}) \quad (4.4.3)$$

拉格朗日函数 $L(\mathbf{X}, \boldsymbol{\lambda})$ 的驻点条件和等式约束条件式(4.4.2)就是最优化解的必要条件。从理论上可以证明,求解拉格朗日函数的最小点并满足等式约束条件(4.4.2)就是原优化问题(4.4.1)和(4.4.2)的解。因此,我们用外点罚函数法求解。其罚函数形式为

$$A(\mathbf{X}, \boldsymbol{\lambda}, \gamma_k) = f(\mathbf{X}) + \sum_{j=1}^l [\lambda_j h_j(\mathbf{X}) + \gamma_k h_j^2(\mathbf{X})] \quad (4.4.4)$$

由上式定义的函数 $A(\mathbf{X}, \boldsymbol{\lambda}, \gamma_k)$ 称为增广拉格朗日函数,显然,式(4.4.4)是外点罚函数的一种修正形式。这种形式的函数有某些引人入胜的特性。第一,如果所有拉格朗日乘子 $\lambda_j=0$, $A(\mathbf{X}, \boldsymbol{\lambda}, \gamma_k)$ 还原为外点罚函数法的惩罚函数 $\varphi(\mathbf{X}, \gamma)$ 。第二,如果选择 $\lambda_j=\lambda_j^*$ (λ_j^* 是满足 Kuhn-Tucker 条件的一组解),对于任意 $\gamma_k > 0$, $A(\mathbf{X}, \boldsymbol{\lambda}^*, \gamma_k)$ 的最优解就是原优化问题(4.4.1)和(4.4.2)的一个真正的最优解。如果我们预先知道 $\boldsymbol{\lambda}^*$,只需要一次无约束最优化问题的求解就可以求得等式约束最优化问题的最优解。第三,我们注意到用古典的外点罚函数法求解,只有当惩罚因子 $\gamma \rightarrow \infty$,才能保证得到精确的最优解,而所有 $\boldsymbol{\lambda}=\boldsymbol{\lambda}^*$,只需要一次无约束最优化问题的求解就可以求得等式约束最优化问题的最优解,因此,ALM法对于有限的惩罚因子,有可能得到等式约束最优化问题的精确可行的最优解。

在实际中, 因为预先不知道 λ^* , 因此, 我们必须设计一种迭代方法, 对于任意初始点 λ_0 (通常取 0 或 1), 能很快地接近最优解 λ^* 。当然, 我们把 λ_j 作为独立变量是适合的, 但是, 这会使设计变量数增加, 因此, 我们宁愿用其他方法。通常建议使用的方法是从 $\lambda_j^0=0$ ($j=1,2,\cdots,l$) 和一个小的惩罚因子 γ 开始。另外, 我们也可以选择 $\lambda_j^0=1$, 对于 $\nabla h_j(\mathbf{X})^T \nabla F(\mathbf{X}) < 0$; 而对于 $\nabla h_j(\mathbf{X})^T \nabla F(\mathbf{X}) > 0$, $\lambda_j^0=-1$ 。后者具有 λ 在数值上从正确方向开始的优点。让 λ_k 和 γ_k 保持为常数, 求 $\min A(\mathbf{X}, \lambda_k, \gamma_k)$, 令 \mathbf{X}_k 为问题的最优解。下一阶段的迭代, 拉格朗日乘子 λ 按下式进行迭代:

$$\lambda_j^{k+1} = \lambda_j^k + 2\gamma_k h_j(\mathbf{X}_k) \quad (j=1,2,\cdots,l) \quad (4.4.5)$$

惩罚因子按下式计算:

$$\gamma_{k+1} = \begin{cases} c\gamma_k & c\gamma_k \leq \gamma_{\max} \\ \gamma_{\max} & c\gamma_k > \gamma_{\max} \end{cases} \quad (4.4.6)$$

其中, γ_{\max} 为设定的惩罚因子的上限。可取 $\gamma_{\max}(\gamma_{\max} > 0)$ 任意常数。但适当地选择较大的 γ_{\max} , 能加快算法的收敛速度。

尽管我们对于拉格朗日乘子的修正公式(4.4.5)的可行性没有证明, 但它是建立在理论基础上的, 对于理论感兴趣的读者可以参考有关的书籍。

关于收敛准则, 通常建议使用 $\lambda_j^{k+1} = \lambda_j^k$ ($j=1,2,\cdots,l$)。不过也可以使用古典罚函数法的收敛准则, 即 $\alpha(\mathbf{X}_{k+1}) < \varepsilon$ 。

例 4.4.1 用增广拉格朗日乘子法求解如下最优化问题:

$$\min f(\mathbf{X}) = x_1^2 + x_2^2$$

满足于

$$h_1(\mathbf{X}) = x_1 + x_2 - 1 = 0$$

解 用拉格朗日乘子法不难求得本题的最优解为 $x_1^* = x_2^* = 0.5$, $\lambda^* = -1$ 。

本题的增广拉格朗日函数为

$$A(\mathbf{X}, \lambda, \gamma) = x_1^2 + x_2^2 + \lambda(x_1 + x_2 - 1) + \gamma(x_1 + x_2 - 1)^2$$

求增广拉格朗日函数 $A(\mathbf{X}, \lambda, \gamma)$ 的梯度

$$\frac{\partial A}{\partial \mathbf{X}} = \begin{pmatrix} \frac{\partial A}{\partial x_1} \\ \frac{\partial A}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 2x_1 + \lambda + 2\gamma(x_1 + x_2 - 1) \\ 2x_2 + \lambda + 2\gamma(x_1 + x_2 - 1) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

解上式, 可得最优点为

$$x_1^* = x_2^* = \frac{2\gamma - \lambda}{4\gamma + 2}$$

显然, 若取 $\lambda = \lambda^* = -1$, 从上式可知, 对于任意惩罚因子 γ , 可得到该最优化问题的最优解为 $x_1^* = x_2^* = 0.5$ 。

现在, 用 ALM 法求解这个问题, 我们考虑如下三种情况。

第一种情况 取 $\lambda_0 = 0$, $\gamma_0 = 1$, $\gamma_{\max} = 1$, 求解无约束最优化问题极小点的前 5 次迭代点的结果如表 4.4.1 所示。从表 4.4.1 中可以看出最优解是从 $h_1(\mathbf{X})$ 的负方向接近于最优解。

第二种情况 取 $\lambda_0 = -2$, $\gamma_0 = 1$, $\gamma_{\max} = 1$, 求解无约束最优化问题极小点的前 5 次迭代点的结果如表 4.4.2 所示。从表 4.4.2 中可以看出最优解是从 $h_1(X)$ 的正方向接近于最优解。

第三种情况 取 $\lambda_0 = 0$, $\gamma_0 = 1$, $c = 10$, $\gamma_{\max} = 100$, 求解无约束最优化问题极小点的前 4 次迭代点的结果如表 4.4.3 所示。从表 4.4.3 中可以看出最优解是从 $h_1(X)$ 的负方向接近于最优解。

表 4.4.1 迭代数据 (一)

| 迭代次数 | λ | γ | $x_1 = x_2$ | $h_1(X)$ |
|------|-----------|----------|-------------|----------|
| 1 | 0.000 | 1 | 0.33333 | -0.33333 |
| 2 | -0.6667 | 1 | 0.44444 | -0.11111 |
| 3 | -0.8889 | 1 | 0.48148 | -0.03704 |
| 4 | -0.9630 | 1 | 0.49381 | -0.01235 |
| 5 | -0.9877 | 1 | 0.49794 | -0.00041 |

表 4.4.2 迭代数据 (二)

| 迭代次数 | λ | γ | $x_1 = x_2$ | $h_1(X)$ |
|------|-----------|----------|-------------|----------|
| 1 | -2.0000 | 1 | 0.66667 | 0.33333 |
| 2 | -1.3333 | 1 | 0.55556 | 0.11111 |
| 3 | -1.1111 | 1 | 0.51852 | 0.03704 |
| 4 | -1.0370 | 1 | 0.50617 | 0.01235 |
| 5 | -1.0123 | 1 | 0.50206 | 0.00041 |

表 4.4.3 迭代数据 (三)

| 迭代次数 | λ | γ | $x_1 = x_2$ | $h_1(X)$ |
|------|-----------|----------|-------------|----------|
| 1 | 0 | 1 | 0.33333 | -0.33333 |
| 2 | -0.6667 | 10 | 0.49206 | -0.01587 |
| 3 | -0.9841 | 100 | 0.49996 | -0.00008 |
| 4 | -0.9999 | 100 | 0.50000 | -0.00000 |

从上例可知, 为了得到最优解, ALM 法并不要求惩罚因子取得很大; 但前几步增大惩罚因子可以改善算法的收敛速度。另一方面, ALM 法从等式约束的一边接近最优解, 从哪一边接近最优解决定于初始拉格朗日乘子 λ_0 的选取。

在下面我们讨论不等式约束的情况中, ALM 法或者是从可行域接近最优解, 或者是从可行域外接近最优解。

4.4.2 不等式约束最优化问题

不等式约束最优化问题为

$$\min f(X) \quad (4.4.7)$$

$$\text{满足于} \quad g_i(\mathbf{X}) \leq 0 \quad (i=1,2,\dots,m) \quad (4.4.8)$$

第一步我们引入松弛变量, 将不等式约束问题转化为等式约束问题

$$g_i(\mathbf{X}) + Z_i^2 = 0 \quad (i=1,2,\dots,m) \quad (4.4.9)$$

用 Z_i^2 而不用 Z_i 是为了数学上处理方便, 因为无论 Z_i 是正数还是负数, Z_i^2 总是正数。在式 (4.4.4) 中, 用 $g_i(\mathbf{X}) + Z_i^2$ 代替 $h_i(\mathbf{X})$ 得到增广拉格朗日函数

$$A(\mathbf{X}, \boldsymbol{\lambda}, Z, \gamma) = f(\mathbf{X}) + \sum_{i=1}^m \left\{ \lambda_i [g_i(\mathbf{X}) + Z_i^2] + \gamma [g_i(\mathbf{X}) + Z_i^2]^2 \right\} \quad (4.4.10)$$

为了建立增广拉格朗日函数, 必须增加 m 个松弛变量 Z_i^2 , 大大增加了设计变量, 然而, 式 (4.4.10) 在数学上等价于下式^[16]:

$$A(\mathbf{X}, \boldsymbol{\lambda}, \gamma) = f(\mathbf{X}) + \sum_{i=1}^m \lambda_i \varphi_i + \gamma \varphi_i^2 \quad (4.4.11)$$

其中

$$\varphi_i = \max \left[g_i(\mathbf{X}), \frac{-\lambda_i}{2\gamma} \right] \quad (4.4.12)$$

同等式约束情况一样, 求解 $A(\mathbf{X}, \boldsymbol{\lambda}, \gamma)$ 。我们假设 $f(\mathbf{X})$ 函数和一阶导数连续, $A(\mathbf{X}, \boldsymbol{\lambda}, \gamma)$ 具有函数和一阶导数连续, 但二阶导数在 $g_i(\mathbf{X}) = \frac{-\lambda_i}{2\gamma}$ 处不连续。

拉格朗日乘子按下式进行迭代:

$$\lambda_i^{k+1} = \lambda_i^k + 2\gamma_k \varphi_i \quad (i=1,2,\dots,m) \quad (4.4.13)$$

4.4.3 一般约束最优化问题

一般约束最优化问题是指既有等式约束又有不等式约束的最优化问题, 即

$$\min f(\mathbf{X}) \quad (\mathbf{X} \in R^n)$$

满足于

$$g_i(\mathbf{X}) \leq 0 \quad (i=1,2,\dots,m)$$

$$h_j(\mathbf{X}) = 0 \quad (j=1,2,\dots,l)$$

我们将等式约束最优化问题和不等式约束最优化问题结合起来, 建立一般约束最优化问题的增广拉格朗日函数

$$A(\mathbf{X}, \boldsymbol{\lambda}, \gamma_k) = f(\mathbf{X}) + \sum_{i=1}^m (\lambda_i \varphi_i + \gamma_k \varphi_i^2) + \sum_{j=1}^l [\lambda_{j+m} h_j(\mathbf{X}) + \gamma_k h_j^2(\mathbf{X})] \quad (4.4.14)$$

式中

$$\varphi_i = \max \left[g_i(\mathbf{X}), \frac{-\lambda_i}{2\gamma_k} \right] \quad (4.4.15)$$

设 \mathbf{X}_k 为优化问题 (4.4.14) 的解, 拉格朗日乘子和惩罚因子的迭代公式为

$$\lambda_i^{k+1} = \lambda_i^k + 2\gamma_k \left\{ \max \left[g_i(\mathbf{X}_k), \frac{-\lambda_i^k}{2\gamma_k} \right] \right\} \quad (i=1,2,\dots,m) \quad (4.4.16)$$

$$\lambda_{j+m}^{k+1} = \lambda_{j+m}^k + 2\gamma_k h_j(\mathbf{X}_k) \quad (j=1,2,\dots,l) \quad (4.4.17)$$

$$\gamma_{k+1} = \begin{cases} c\gamma_k & c\gamma_k \leq \gamma_{\max} \\ \gamma_{\max} & c\gamma_k > \gamma_{\max} \end{cases} \quad (4.4.18)$$

增广拉格朗日乘子法程序框图如图 4.4.1 所示。

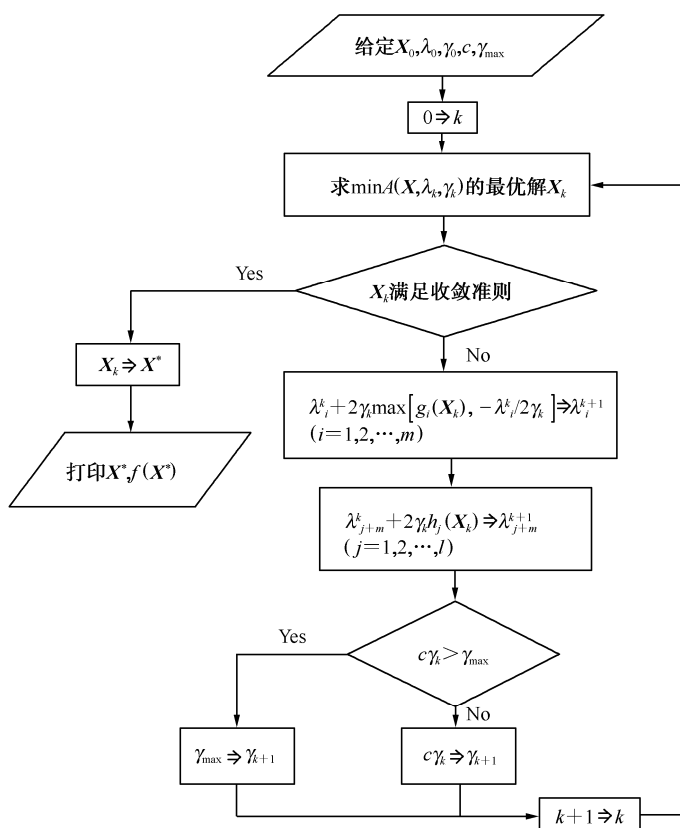


图 4.4.1 增广拉格朗日乘子法程序框图

增广拉格朗日乘子法具有如下优点：

- (1) 方法不依赖于惩罚因子的值，没有必要取 $\gamma_k \rightarrow \infty$ ；
- (2) 求得的最优解 \mathbf{X}^* 有可能精确满足约束条件 $g_i(\mathbf{X}^*) \leq 0$ 和 $h_j(\mathbf{X}^*) = 0$ ；
- (3) 修改拉格朗日乘子能加速优化迭代过程；
- (4) 初始点既可以是可行点，也可以是非可行点；
- (5) 在 \mathbf{X}^* 处，自动地求得起作用的不等式约束条件集的拉格朗日乘子 λ_i^* 。

第 5 章 多目标函数的优化设计方法^[6, 17]

5.1 引言

在最优化设计中, 某个设计方案的好坏仅涉及一项设计指标, 称它为单目标函数的优化设计问题。对于这种优化问题, 应用前面介绍的最优化方法就可以直接求得最优设计方案。然而, 在许多实际问题中, 对于一个设计方案往往期望几项设计指标同时达到最优解。例如, 设计飞剪机剪切机构时 (见第 8 章 8.1 节), 第一项设计指标是: 在剪切过程中, 要求剪切刀刃在水平方向上的速度 V_x 尽量均匀, 并且等于或略大于轧制钢坯的运行速度。第二项设计指标是: 在剪切过程中, 为了减少两个剪切刀刃的阻力和防止刀刃相互干涉, 要求在剪切区域内两个剪切刀刃垂直于轧制钢坯。在优化设计中, 这种同时要求两项或两项以上设计指标达到最优值的问题称为多目标函数的优化问题。

多目标函数的优化设计问题的一般形式为

$$\min[f_1(X), f_2(X), \dots, f_p(X)]^T \quad (p \geq 2, X \in R^n)$$

$$\begin{aligned} \text{满足于} \quad & g_i(X) \leq 0 \quad (i=1, 2, \dots, m) \\ & h_j(X) = 0 \quad (j=1, 2, \dots, l) \end{aligned} \quad (5.1.1)$$

令可行解集 $D = \{X: g_i(X) \leq 0 \quad (i=1, 2, \dots, m); h_j(X) = 0 \quad (j=1, 2, \dots, l); X \in R^n\}$ 和 $F(X) = [f_1(X), f_2(X), \dots, f_p(X)]^T$, 多目标规划可简化为

$$\min F(X) \quad (X \in D) \quad (5.1.2)$$

在求解多目标函数的优化设计中, 同时使几个目标函数达到最优解, 一般是不可能的。在求解过程中, 使一个目标函数达到最优解, 往往会引起另一个或几个目标函数变坏。

求解多目标函数最基本方法为评价函数法。评价函数法的基本思想是: 借助于几何或应用中的直观背景, 构造一个评价函数, 从而将多目标优化问题转化为单目标函数优化问题。然后, 利用单目标函数优化的求解方法求出最优解, 并把这种最优解作为多目标规划的最优解。

5.2 多目标函数优化设计的求解方法

5.2.1 理想点法

在多目标函数优化中, 先求解 p 个单目标函数的最优解:

$$\min f_i(X) \quad (i=1, 2, \dots, p; X \in D) \quad (5.2.1)$$

设其最优值为 f_i^* , 我们称 $F^* = (f_1^*, f_2^*, \dots, f_p^*)^T$ 为值域中的一个理想点。因为一般很难达到理

想点, 我们期望在某种度量下, 寻求值域中距离 F^* 最近的 F 作为近似解。

用下式作为一种尺度:

$$\varphi(Z) = \sqrt{\sum_{i=1}^p (z_i - f_i^*)^2} \quad (5.2.2)$$

式中, $Z = (z_1, z_2, \dots, z_p)^T$ 值域中的一个点。

则多目标函数的优化问题可转化为如下单目标函数的优化问题:

$$\min \varphi[F(X)] = \sqrt{\sum_{i=1}^p [f_i(X) - f_i^*]^2} \quad (X \in D) \quad (5.2.3)$$

5.2.2 线性加权法

在多目标函数优化问题中, 人们总希望对那些相对重要的目标函数给予较大的权系数。基于这个事实, 我们可以构造如下评价函数:

$$\varphi[F(X)] = \sum_{i=1}^p W_i f_i(X) \quad (5.2.4)$$

其中, W_i 称为加权系数, $W_i > 0$ 。

则多目标函数的优化问题可转化为如下单目标函数的优化问题:

$$\min \varphi[F(X)] = \sum_{i=1}^p W_i f_i(X) \quad (X \in D) \quad (5.2.5)$$

加权系数 W_i 的选取有两个作用, 其一是通过适当选取 W_i , 使各分目标函数值在数量级上大体相等; 其二是通过适当选取 W_i , 来体现各分目标函数在设计中的重要程度, 突出主要的设计指标。

5.2.3 乘除法

若多目标函数优化问题中, p 个分目标函数分为两类, 第一类是有 s 个分目标函数希望函数值越小越好, 这类目标函数有耗材料、成本、工时和重量等; 第二类是有 t 个分目标函数希望函数值越大越好, 这类目标函数有产品的产量、效率、利润和承载能力等。对于这种情况, 可取如下统一的目标函数:

$$\min \varphi[F(X)] = \frac{\sum_{i=1}^s W_i f_i(X)}{\sum_{j=s+1}^p W_j f_j(X)} \quad (5.2.6)$$

式中, W_i 称为加权系数, $W_i > 0$ 。

值得注意的是: 当 $\sum_{j=s+1}^p W_j f_j(X) = 0$ 或 ≈ 0 时, 乘除法将无法计算。

5.2.4 极大极小法

在决策时,采取保守策略是稳妥的,即在最坏的情况下,寻求最好的结果。按照这种思路,多目标函数优化问题可转化为如下统一的目标函数:

$$\varphi[F(X)] = \max_{1 \leq i \leq p} f_i(X) \quad (5.2.7)$$

然后,求解

$$\min \varphi[F(X)] = \min_{X \in D} \max_{1 \leq i \leq p} f_i(X) \quad (5.2.8)$$

并将它的最优解 \mathbf{X}^* 作为多目标函数优化问题的最优解。

第 6 章 遗传算法简介^[18~20]

6.1 引言

遗传算法是基于达尔文的自然选择原理、自然遗传机制、优胜劣汰和适者生存的生物遗传和进化的规律而形成的一种随机搜索优化算法，它由美国 Michigan 大学 John Holland 教授于 1975 年首先提出。达尔文在 1858 年用自然选择来解释物种的起源和生物的进化，其自然选择学说包括以下三个方面内容：

（1）遗传 遗传是生物的普遍特征，“种瓜得瓜，种豆得豆”，亲代把生物信息交给子代，子代按照所得信息而发育、分化，因而子代和亲代具有相同和相似的性状。生物有了这个特征，物种才能稳定存在。

（2）变异 亲代和子代之间以及子代的不同个体之间总有些差异，这种现象称为变异。变异是随机发生的，变异的选择积累是生命多样性的根源。

（3）生存斗争和适者生存 自然选择来自繁殖过剩和生存斗争。由于弱肉强食的生存斗争不断地进行，其结果是适者生存。适应性强的个体被保留下来，适应性弱的个体被淘汰，通过一代代的生存环境的选择作用，物种特性被定向朝这一方向积累，于是物种的性状与原先的祖先不同，演变为新的物种。这种自然选择过程是一个长期的、缓慢的和连续的过程。

6.2 遗传算法及其特点

6.2.1 遗传算法的基本思路及其特点

遗传算法的基本思路是从代表问题可能潜在解集的一个种群开始的，而一个种群通过基因编码的一定数目的个体组成。每个个体实际上是染色体带有特征的实体，它是遗传物质的主要载体，由多个遗传基因组成。染色体内部表现是某种基因的组合，它决定了个体的形状的外部表现。从表现型（即由染色体决定性状的外部表现）到基因型的映射需要进行编码工作。由于仿照基因编码的工作非常复杂，因此，我们进行简化编码，如二进制编码。初始种群产生之后，按照适者生存和优胜劣汰的原理，逐代演化产生出越来越好的近似解。在每一代中根据个体的适应度大小挑选个体，并借助自然遗传学的遗传算子进行组合交叉和变异，产生出代表新的解集的种群。这个过程将导致像自然进化一样的后代种群比前代更加适应于环境，末代种群中的最好个体经过解码，可以作为优化问题的近似最优解。

利用遗传算法解最优化问题，首先，应对可行域中的点进行编码，一般采用二进制编码；

然后,在可行域中随机挑选一些编码作为进化起点的第一代编码组,并计算每个解的目标函数值,也就是编码的适应度。接着就像自然界中生物进化一样,利用选择机制从编码组中随机挑选编码作为繁殖过程前的编码样本。选择机制应保证适应度较高的解保留较多的样本;而适应度较低的解则保留较少的样本,甚至被淘汰。在繁殖过程中,遗传算法提供了交叉和变异两种算子对挑选后的样本进行交换。交叉算子交换随机挑选的两个编码的某些位,变异算子则直接对一个编码中的随机挑选的某一位进行反转。这样通过选择和繁殖就产生了下一代编码组。重复上述选择和繁殖过程,直到满足结束条件为止。进化过程最后一代中的最优解就是用遗传算法解最优化问题所得到的最终结果。

从以上介绍可以看出,遗传算法具有下述特点:

- (1) 遗传算法不是直接处理优化问题变量本身的实际值,而是以优化问题变量的编码为运算对象。
- (2) 遗传算法是从优化问题解的编码组开始搜索的,而不是从单个解开始搜索的。
- (3) 遗传算法不要求目标函数连续,更不要求目标函数可微。
- (4) 遗传算法使用的选择、交叉、变异这三个算子都是随机操作,而不是确定的规则。

6.2.2 遗传算法的基本步骤

我们习惯上把 John Holland 教授 1975 年提出的遗传算法称为传统的遗传算法,它由如下五个重要的环节组成。

1. 编码和初始群体的生成

遗传算法在进行搜索之前,先将解空间的解数据表示成表达空间(也称遗传空间)的基因型串结构数据。然后,随机产生 M 个初始串结构数据,每个串结构数据称为一个个体, M 个个体构成了一个群体。遗传算法以这 M 个串结构数据作为初始点开始迭代。

2. 验证收敛准则

以验证算法的收敛准则是否满足来控制算法是否结束,可以采用最大迭代次数或者与适应度有关的准则来判断是否终止遗传算法运算。

3. 适应度评估检测和选择

适应度函数表明个体或解的优劣性,在程序开始时应该评价个体的适应度,以便进行比较。不同的问题,适应度函数的定义方式是不同的。根据适应度的好坏,进行选择。选择的目的是为了从当前群体中选出优良的个体,使它们有机会作为父代为下一代繁殖子孙。进行选择的原则是适应度强的个体为下一代贡献一个或多个后代的概率大。遗传算法通过选择过程体现这一思想,选择实现了达尔文的适者生存原则。

4. 交叉

按照交叉概率 P_c 进行交叉。交叉是遗传算法中最主要的遗传算子,通过交叉运算可以得到新一代个体,新个体组合了其父辈个体的特性。

交叉体现了信息交换的思想，可以选定一个点对两个编码串进行互换，也可以随机选取几个点进行交叉运算。交叉概率如果太大，种群更新快，但是，高适应度的个体很容易被淹没；交叉概率太小，会使搜索停滞不前。

5. 变异

按照变异概率 P_m 进行变异。首先，在群体中随机选择一个个体，对于选中的个体以一定的概率随机地改变串结构数据中某一位的值。同生物界一样，遗传算法中变异发生的概率很低。变异为新个体的产生提供了机会。

综上所述，标准遗传算法的主要步骤如下：

- (1) 随机产生一组初始个体构成的初始群体。
- (2) 计算群体中个体的适应度。
- (3) 判断算法收敛准则是否满足，若满足就输出搜索结果并停止运算，否则转下一步。
- (4) 选择运算，按优胜劣汰规律执行复制操作。
- (5) 交叉运算，按一定的方式进行交叉操作。
- (6) 变异运算，按一定的规律执行变异操作。
- (7) 转到步骤 (2)。

遗传算法程序框图如图 6.2.1 所示。

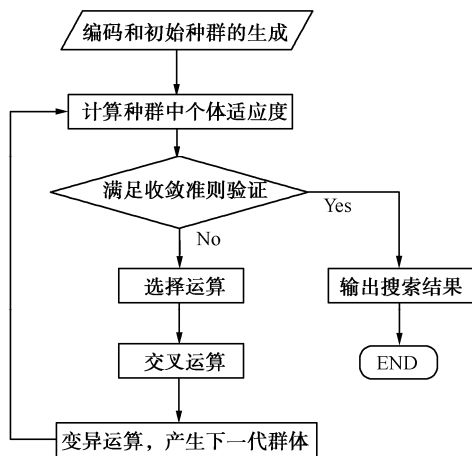


图 6.2.1 遗传算法程序框图

6.3 遗传算法的实现方法

6.3.1 编码方法

遗传算法不是对求解问题的变量直接进行操作，而是通过对个体编码进行交叉和变异的进化运算过程，不断搜索出适用度高的新个体，最终寻求问题的最优解或近似最优解。遗传

算法不能直接处理优化问题的空间参数，必须把它们转化成遗传算法表达空间的基因链码的形式，这一转换操作称为编码。

通过编码之后，问题的解转而由某种基因链码形式表示，我们称该基因链码的所有个体构成了表达空间。因此，编码问题实际是从问题空间到表达空间的映射问题。设计编码策略时，应考虑以下三个问题：

(1) 完备性 对问题空间的任何一个点有表达空间的一个点与之对应，即问题空间的有可能解都能表示为所设计的基因编码形式。

(2) 健全性 对于表达空间中的任何一个点都有问题空间中的一个点与之对应，即任何一个基因编码都对应于一个可行解。

(3) 非冗余性 问题空间和表达空间一一对应。

下面，介绍几种常用的编码方法。

1. 二进制编码

二进制编码是将原问题的解映射成 0 和 1 组成的编码串的表达空间，结果再通过解码过程还原成其解空间的解，然后，再进行适应度的计算。

由于很多最优化问题都可以用二进制编码来应用遗传算法，同时二进制编码表达的模式最多，所以，二进制编码方法是遗传算法中最常用的一种编码方法。它具有如下优点：

(1) 编码、解码操作简单易行。

(2) 交叉、变异等遗传操作便于实现。

(3) 符合最小字符集编码原理。

(4) 便于用模式定理进行分析，因为，模式定理（模式定理见参考文献[19]）就是以二进制为基础的。

二进制编码符号的长度与问题所要求的精度有关。假设某一参数的取值范围是 $[U_{\min}, U_{\max}]$ ，用长度为 l 的二进制编码串来表示该参数，则总共能产生 2^l 种不同的编码。若使参数编码时的对应关系如下：

$$\begin{array}{rcl}
 00000 \cdots 00000 & = & 0 \quad \rightarrow U_{\min} \\
 00000 \cdots 00001 & = & 1 \quad \rightarrow U_{\min} + \delta \\
 00000 \cdots 00010 & = & 2 \quad \rightarrow U_{\min} + 2\delta \\
 \vdots & & \vdots \\
 11111 \cdots 11111 & = & 2^l - 1 \quad \rightarrow U_{\max}
 \end{array}$$

二进制编码的编码精度 δ 为

$$\delta = \frac{U_{\max} - U_{\min}}{2^l - 1} \quad (6.3.1)$$

假设某个个体的二进制编码是

$$S: b_l b_{l-1} b_{l-2} \cdots b_2 b_1 \quad (6.3.2)$$

则对应的解码公式为

$$x = U_{\min} + \left(\sum_{i=1}^l b_i \times 2^{i-1} \right) \frac{U_{\max} - U_{\min}}{2^l - 1} \quad (6.3.3)$$

例 6.3.1 对于 $x \in [0, 31]$, 若用 5 位长的二进制编码来表示该参数, 则 $S=01101$ 对应于问题空间中 $x=13$, 按式 (6.3.1) 计算, 编码精度 $\delta=1$ 。若要精确到小数的后 1 位, 需要 9 位长的二进制编码。

二进制编码有如下缺点:

(1) 相邻整数的二进制编码可能具有较大 Hamming 距离, 例如 15 和 16 的二进制表示为 01111 和 1000, 因此, 算法要从 15 改进到 16 时必须改变所有的位。二进制编码的这一缺点有时被称为 Hamming 悬崖 (Hamming Cliffs)。

(2) 二进制编码时, 一般要先给出求解的精度以确定串长, 而一旦精度确定后, 就很难在算法执行过程中进行调整, 从而使算法缺乏微调功能。若在算法一开始就选取较高的精度, 那么, 串长较大, 这样就会降低算法的效率。

(3) 在求解高维优化问题时, 二进制编码串长非常大, 从而使得算法的搜索效率很低。

2. 格雷 (Gray) 编码

格雷编码是将二进制编码通过一个变换得到的编码。设有二进制编码串为 $B=b_l b_{l-1} b_{l-2} \cdots b_2 b_1$, 对应的格雷编码为 $G=g_l g_{l-1} g_{l-2} \cdots g_2 g_1$, 按以下公式将二进制编码转化为格雷编码:

$$\begin{cases} g_l = b_l \\ g_i = b_{i+1} \oplus b_i \quad (i=l-1, l-2, \cdots, 2, 1) \end{cases} \quad (6.3.4)$$

式中, 符号 \oplus 表示异或运算符, 如 $1 \oplus 0=1$, $1 \oplus 1=0$ 。

格雷编码转化为二进制编码的公式为

$$\begin{cases} b_l = g_l \\ b_i = b_{i+1} \oplus g_i \quad (i=l-1, l-2, \cdots, 2, 1) \end{cases} \quad (6.3.5)$$

格雷编码的目的是克服二进制编码的 Hamming 悬崖, 改善连续函数优化问题的局部搜索能力。格雷编码在连续的两位整数所对应的编码值之间仅有一个码位不相同, 其余码位完全相同。表 6.3.1 所示为十进制数 0~15 对应的二进制编码和格雷编码。

例 6.3.2 对于 $x \in [0, 31]$, $x=19$ 的二进制编码为 10011, 转化为格雷编码为 11010; $x=20$ 的格雷编码为 11110, 二进制编码为 10100。

表 6.3.1 二进制编码和格雷编码

| 十进制数 | 二进制编码 | 格雷编码 |
|------|---------|---------|
| 0 | 0 0 0 0 | 0 0 0 0 |
| 1 | 0 0 0 1 | 0 0 0 1 |
| 2 | 0 0 1 0 | 0 0 1 1 |
| 3 | 0 0 1 1 | 0 0 1 0 |
| 4 | 0 1 0 0 | 0 1 1 0 |

续表

| 十进制数 | 二进制编码 | 格雷编码 |
|------|---------|---------|
| 5 | 0 1 0 1 | 0 1 1 1 |
| 6 | 0 1 1 0 | 0 1 0 1 |
| 7 | 0 1 1 1 | 0 1 0 0 |
| 8 | 1 0 0 0 | 1 1 0 0 |
| 9 | 1 0 0 1 | 1 1 0 1 |
| 10 | 1 0 1 0 | 1 1 1 1 |
| 11 | 1 0 1 1 | 1 1 1 0 |
| 12 | 1 1 0 0 | 1 0 1 0 |
| 13 | 1 1 0 1 | 1 0 1 1 |
| 14 | 1 1 1 0 | 1 0 0 1 |
| 15 | 1 1 1 1 | 1 0 0 0 |

3. 实数编码

为了克服二进制编码的缺点，当问题的变量是实向量时，直接采用十进制数进行编码。直接采用十进制的实数编码是连续参数优化问题的直接描述，不存在编码和解码过程。

在实数编码中常有浮点数出现，一般将个体的每个基因用某一范围内的一个浮点数来表示。个体的编码长度等于决策变量的个数。

4. 多参数映射编码

通常优化问题的求解中往往会碰到多参数优化问题，例如，函数 $f(x,y)=x^2y^2$ ，而 $x\in[0,63]$ 、 $y\in[0,63]$ 的优化问题。可以采用如下编码方法：将 x 用 8 位的二进制编码放在前面， y 用 8 位的二进制编码放在后面，构成 16 位二进制编码。

一般来说，每个参数可以用不同的编码长度或编码精度，也可以具有不同的上下限。

5. 动态编码方法

动态编码方法是当算法收敛到局部极值时增加搜索的精度，增加精度的办法是在保持串长不变的前提下减小搜索区域。

6.3.2 适应度函数

在遗传算法中，用适应度评估个体或解的优劣，这个评估个体适应度的函数称为适应度函数。适应度较高的个体遗传到下一代的概率大，而适应度较低的个体遗传到下一代的概率小。为了用遗传算法求解最优化问题，应该建立遗传算法中适应度函数与最优化问题目标函数的某种对应关系，下面，讨论这个问题。

1. 目标函数映射成适应度函数

对于函数的优化问题, 必须将优化问题的目标函数 $f(x)$ 与遗传算法中个体的适应度函数 $fit(S)$ 建立一定的映射关系, S 为解空间变量 X 对应的遗传算法表达空间的基因链码。由于遗传算法中要对个体的适应度比较排序, 并在此基础上确定选择概率, 所以适应度函数 $fit(S)$ 应取正数。同时, 为了使遗传算法的最优解能代表优化问题的最优解, 要求在优化过程中目标函数 $f(X)$ 的变化方向 [如求 $\min f(X)$ 或 $\max f(X)$] 应与群体进化过程中的适应度函数 $fit(S)$ 的变化方向一致。

对于求极小化的目标函数 $\min f(X)$, 可取如下适应度函数 $fit(S)$:

$$fit(S) = \begin{cases} C_{\max} - f(X) & C_{\max} - f(X) > 0 \\ 0 & C_{\max} - f(X) \leq 0 \end{cases} \quad (6.3.6)$$

对于求极大化的目标函数 $\max f(X)$, 可取如下适应度函数 $fit(S)$:

$$fit(S) = \begin{cases} f(X) + C_{\min} & f(X) + C_{\min} > 0 \\ 0 & f(X) + C_{\min} \leq 0 \end{cases} \quad (6.3.7)$$

式中, C_{\min} 和 C_{\max} 为可调参数。 C_{\max} 可以是一个合适的输入值, 也可以用迄今为止进化进程中 $f(X)$ 的最大值, 或者 $f(X)$ 的最大值。同样, C_{\min} 可以是一个合适的输入值, 或者当前代或前 K 代 $f(X)$ 的最小值, 或者 $f(X)$ 的最小值。

2. 适应度函数调整

在遗传算法的初始阶段, 群体中可能存在少数适应度很高的个体, 若使用基于适应度的概率选择方法, 这些个体就会被大量繁殖, 从而在群体中占有很大比重。而适应度很低的个体在群体中过早地被淘汰, 造成群体多样性急剧下降。这样可能导致群体早熟提前收敛, 或者将遗传算法搜索导向局部极值点。

在遗传算法的迭代的后期阶段, 群体中所有个体的适应度可能接近于群体中最佳个体的适应度。这时, 所有个体无竞争力, 它们都会有相接近的概率遗传到下一代。这将导致无法对重点区域进行重点搜索。

基于以上理由, 在遗传算法的不同阶段, 有必要对适应度函数进行适当的扩大或缩小。这种适应度函数的缩放称为适应度函数调整。目前, 常用的个体适应度函数调整有线性调整和幂调整等。

(1) 线性调整

设原适应度函数为 $fit(S)$, 调整后的适应度函数为 $Fit(S)$, 则线性调整可用下式表示:

$$Fit(S) = a \cdot fit(S) + b \quad (6.3.8)$$

式中, 系数 a 和 b 为待定常数, 它们满足如下两个条件:

- ① 原适应度的平均值 fit_{avg} 等于调整后适应度的平均值 Fit_{avg} 。
- ② 调整后适应度的最大值 Fit_{max} 要等于原适应度平均值的指定的 C 倍, 即

$$Fit_{\text{max}} = C \cdot fit_{\text{avg}} \quad (6.3.9)$$

其中, C 为最优个体所期望的复制数。

实验表明, 对于一个 50~100 个体的群体, 一般取 $C=1.2\sim 2$ 。此条件是保证群体中最好

的个体能够复制 C 倍到新一代群体中。

由式 (6.3.8) 可得

$$\begin{cases} Fit_{\max} = a \cdot fit_{\max} + b \\ Fit_{\text{avg}} = a \cdot fit_{\text{avg}} + b \end{cases} \quad (6.3.10)$$

由系数 a 和 b 满足的两个条件, 解上式可得

$$\begin{cases} a = \frac{C-1}{Fit_{\max} - fit_{\text{avg}}} fit_{\text{avg}} \\ b = \frac{fit_{\max} - C \cdot fit_{\text{avg}}}{Fit_{\max} - fit_{\text{avg}}} fit_{\text{avg}} \end{cases} \quad (6.3.11)$$

在搜索过程后期, C 有可能对原适应度函数进行了过分收缩, 使少数异常个体调整后适应度为负数, 这将给后面的处理带来不便。解决问题的方法可修改系数 C , 使调整后的适应度大于零。将式 (6.3.11) 代入式 (6.3.8), 经推导可得

$$C \leq \frac{fit_{\max} - fit_{\min}}{fit_{\text{avg}} - fit_{\min}} \quad (6.3.12)$$

(2) 幂调整

幂调整可用下式计算:

$$Fit(S) = fit(S)^k \quad (6.3.13)$$

上式为调整后适应度是原适应度的某个指定乘幂。幂指数 k 与所求的问题有关, 并且在算法执行过程中需要不断对其进行修订, 才能满足一定的缩放要求。

6.3.3 遗传算子

遗传算法中包括三个基本遗传算子: 选择、交叉和变异。

1. 选择算子

选择算子 (或称复制算子) 是对群体中的个体按照优胜劣汰的方式选取, 并遗传到下一代群体的运算操作, 它是建立在群体中个体的适应度评估基础上的。适应度较高的个体被遗传到下一代群体中的概率较大, 适应度较低的个体被遗传到下一代群体中的概率较小。下面介绍经常用到的几种选择方法。

(1) 适应度比例方法

适应度比例方法又称轮盘赌选择方法, 是目前遗传算法中最基础的、也是最常用的选择方法。在该方法中, 每个个体的选择概率与其适应度值成比例。

设群体大小为 M , 个体 i 的适应度为 fit_i , 则个体 i 被选中的概率为

$$P_i = \frac{fit_i}{\sum_{i=1}^M fit_i} \quad (i=1, 2, \dots, M) \quad (6.3.14)$$

概率 P_i 反映了个体 i 的适应度在整个个体的适应度总和中所占的比例, 可以用轮盘赌中转盘来直观地理解该选择方法。首先, 将一个圆盘 (或转盘) 分为 M 个扇形, 第 i 个扇形的中心角为 $2\pi P_i$ 。其次, 在进行选择时, 假想转动一下圆盘, 若参照点落入到第 i 个扇形内, 则选择个体 i 。显然, 适应度比例方法类似于轮盘赌方法, 扇形的中心角越大, 参照点落入该扇形内的概率就越大, 即个体的适应度值越大, 它被选择的机会也越多。表 6.3.2 给出了采用适应度比例方法的个体的适应度与选择概率的例子。

表 6.3.2 个体的适应度与选择概率的关系

| 个体 | 适应度 | 选择概率 |
|----|------|------|
| 1 | 1.0 | 0.07 |
| 2 | 3.0 | 0.22 |
| 3 | 2.5 | 0.18 |
| 4 | 0.8 | 0.06 |
| 5 | 1.8 | 0.13 |
| 6 | 1.2 | 0.09 |
| 7 | 0.9 | 0.07 |
| 8 | 0.1 | 0.01 |
| 9 | 0.9 | 0.07 |
| 10 | 1.4 | 0.10 |
| 累计 | 13.6 | 1 |

(2) 最佳个体保存方法

该方法的思想是把群体中适应度最高的个体不进行交叉而直接复制到下一代中, 当然, 这样做的前提是下一代中不存在该个体。

采用这种方法的优点是进化过程中某一代的最优解可不被交叉或变异破坏。但是, 这也隐含一种危机, 即局部最优个体的遗传基因会急剧增加而使进化有可能陷于局部解。也就是说, 该方法的全局搜索能力差。它适应于单峰 (或单谷) 目标函数的优化问题, 而不适应于多峰 (或多谷) 目标函数的优化问题。所以, 该方法一般与其他选择方法结合使用。

(3) 期望值方法

当个体数不大时, 适应度比例选择方法有时因为随机数, 可能出现不正确地反映个体的选择, 即出现统计误差, 也就是说, 适应度高的个体有可能被淘汰, 适应度低的个体有可能被选择。为了克服这种缺点, 期望值方法的基本思想是根据每个个体在下一代群体中的生存期望值来进行随机运算。期望值方法步骤如下:

① 计算群体中每个个体在下一代中生存的期望数目 N_i , 即

$$N_i = \frac{fit_i}{fit_{avg}} = \frac{M \cdot fit_i}{\sum_{j=1}^M fit_j} \quad (i=1, 2, \dots, M) \quad (6.3.15)$$

② 若某个个体被选中并要参与配对和交叉, 则它在下一代中的生存的期望数目减去 0.5, 若不参与配对和交叉, 则该个体的生存的期望数目减去 1。

③ 随着选择过程的进行，若一个个体的期望值小于零，则该个体不参与选择。

(4) 排序选择方法

所谓排序选择方法是指在计算出每个个体的适应度后，根据适应度大小在群体中对个体进行排序，然后，把事先设计好的概率表按序分配给个体，作为各自的选择概率。表 6.3.3 给出了排序选择方法中个体的适应度与选择概率的关系的一个例子。从表中可以看出，所有个体按适应度大小降序排列，选择概率与个体的适应度无直接关系，而仅与序号有关。

表 6.3.3 个体的序号与选择概率的关系

| 序号 | 个体 | 适应度 | 选择概率 |
|----|----|-----|------|
| 1 | 3 | 3.0 | 0.35 |
| 2 | 1 | 2.5 | 0.20 |
| 3 | 7 | 2.5 | 0.15 |
| 4 | 5 | 2.1 | 0.10 |
| 5 | 10 | 1.8 | 0.06 |
| 6 | 8 | 1.3 | 0.05 |
| 7 | 4 | 1.2 | 0.04 |
| 8 | 2 | 1.0 | 0.03 |
| 9 | 9 | 0.9 | 0.02 |
| 10 | 6 | 0.8 | 0.00 |

2. 交叉算子

在遗传算法中，使用交叉算子产生后代个体来模仿两个同源染色体通过交配重组形成新染色体的生物进化过程。在遗传算法中，交叉算子的作用非常重要。一方面，它使得在原来的群体中优良个体的特性能够在一定程度上保持；另一方面，它使得算法能够探索新的基因空间，从而使新的群体中的个体具有多样性。下面，介绍经常使用的交叉算子。

(1) 单点交叉

单点交叉又称简单交叉，其具体操作是在个体基因串中随机设定一个交叉点，实行交叉时，两个配对父辈个体的部分染色体在设定点相互交换，并生成新的个体。下面，给出单点交叉的例子。

$$\begin{array}{lcl}
 \text{父辈个体 } A & 1101 \mid 001 & \\
 \text{父辈个体 } B & 1001 \mid 011 & \\
 \hline
 & \xrightarrow{\text{单点交叉}} & \\
 \text{后代个体 } A' & 1101011 & \\
 \text{后代个体 } B' & 1001001 &
 \end{array}$$

在以上例子中，配对父辈个体 A 和个体 B 为 7 位二进制编码，交叉点设在第 4 个和第 5 个基因座之间。交叉时，该交叉点后的两个父辈个体的部分编码相互交换。父辈个体 A 第 1 个到第 4 个基因与父辈个体 B 第 5 个到第 7 个基因组成一个后代个体 A' ，父辈个体 B 第 1 个到第 4 个基因与父辈个体 A 第 5 个到第 7 个基因组成一个后代个体 B' 。交叉点是随机设定的，当染色体长度为 l 时，有 $l-1$ 个可能的交叉点，所以，单点交叉可能实现 $l-1$ 个不同的交叉结果。

(2) 两点交叉和多点交叉

两点交叉是在个体编码串中随机设定两个交叉点，然后，将两个父辈配对个体的中间部

分基因进行交换。两点交叉的例子如下：

$$\begin{array}{lcl} \text{父辈个体 } A & 11 \mid 010 \mid 01 & \xrightarrow{\text{两点交叉}} & 1111001 & \text{后代个体 } A' \\ \text{父辈个体 } B & 10 \mid 110 \mid 11 & & 1001011 & \text{后代个体 } B' \end{array}$$

在以上例子中，配对父辈个体 A 和个体 B 为 7 位二进制编码，第 1 个交叉点设在第 2 个和第 3 个基因座之间，第 2 个交叉点设在第 5 个和第 6 个基因座之间。交叉时，将父辈配对个体 A 和个体 B 在第 1 个交叉点和第 2 个交叉点之间部分基因进行相互交换，分别生成了后代个体 A' 和个体 B' 。

多点交叉又称为广义交叉，它是在个体编码串中随机设置多个交叉点，然后，进行基因交换。

三点交叉的例子如下：

$$\begin{array}{lcl} \text{父辈个体 } A & aa \mid aa \mid aaa \mid aaa & \xrightarrow{\text{三点交叉}} & aabbaaabb & \text{后代个体 } A' \\ \text{父辈个体 } B & bb \mid bb \mid bbb \mid bbb & & bbaabbbba & \text{后代个体 } B' \end{array}$$

一般来讲，多点交叉不经常被采用，这是因为随着交叉点的增加，个体的结构被破坏的可能性也逐渐增大，有可能破坏一些好的模式，很难有效地保存一些较好的模式，从而影响遗传算法的性能。

(3) 均匀交叉

均匀交叉则是依照概率交换两个父辈个体基因串的每一位。其过程是：先随机地产生一个与父辈个体基因串具有相同长度的二进制串，其中，0 表示不交换，1 表示交换。这个二进制串称为交叉模板。然后，根据该模板对两个父辈基因串进行交叉，得到的两个新基因串即为后代新个体。例如，模板串为 001101011100，均匀交叉为：

$$\begin{array}{lcl} \text{父辈个体 } A & 110010111000 & \xrightarrow{\text{通过模板串}} & 111011101000 & \text{后代个体 } A' \\ \text{父辈个体 } B & 101011101011 & & 100010111011 & \text{后代个体 } B' \end{array}$$

由于均匀交叉在交换位时并不考虑其所在位置，破坏模式的概率较大。但另一方面它能搜索到一些基于点交叉方法无法搜索到的模式。

(4) 算术交叉

对于实数编码的个体可采用算术交叉的方法。它是由两个父辈个体的线性组合而产生两个后代个体。若两个父辈个体 X_A^t 和 X_B^t 之间进行算术交叉，则交叉运算后所产生的两个后代个体 X_A^{t+1} 和 X_B^{t+1} 为

$$\begin{cases} X_A^{t+1} = \alpha X_B^t + (1 - \alpha) X_A^t \\ X_B^{t+1} = \alpha X_A^t + (1 - \alpha) X_B^t \end{cases} \quad (6.3.16)$$

式中， α 为一参数。它可以是一个常数，此时所进行的交叉运算称为均匀算术交叉；它可以是一个由进化代数所决定的变量，此时所进行的交叉运算称为非均匀算术交叉。

在很多应用中，交叉算子是以一定的概率实现的，这一概率称为交叉概率 P_c 。

3. 变异算子

变异操作模仿自然界生物体进化中染色体上某位基因发生的突变现象，从而改变染色体的结构和物理性状。在遗传算法中，用变异算子来产生出新的个体。变异算子的基本内容是对群体中的个体编码串的某些基因位置上的基因值作变动。对于二进制的编码串而言，变异操作就是以变异概率 P_m 把某个基因位置上的基因值取反，即 $0 \rightarrow 1$ 或 $1 \rightarrow 0$ 。对于有浮点数编

码的个体,若某一变异点的基因值的取值范围为 $[U_{\min}, U_{\max}]$,变异操作就是用该范围内的一个随机数去替换原基因值。

变异算子操作的基本步骤如下:

- ① 在群体中所有个体的编码串范围内,随机地确定基因位置。
- ② 以事先设定的变异概率 P_m 来对这些基因位置的基因值进行变换。

在遗传算法中,交叉算子因其全局搜索能力而作为主要算子,变异算子因局部搜索能力而作为辅助算子。遗传算法通过交叉和变异这一对相互配合和相互竞争的操作,而使其具备兼顾全局和局部的搜索能力。所谓相互配合是指群体在进化中陷于搜索空间某个超平面而仅靠交叉不能摆脱时,通过变异操作可能有助于这种摆脱。所谓相互竞争,是指当前通过交叉已形成所期望的基因模式时,变异操作有时可能破坏这些基因模式。

遗传算法中使用变异算子使得遗传算法有局部的搜索能力,而且可以使遗传算法维持群体的多样性。下面,介绍二进制变异和实值变异算子。

(1) 二进制变异

对群体中的基因链码随机挑选 c 个基因位置,并对这些基因位置的基因值以变异概率取反,即 $0 \rightarrow 1$ 或 $1 \rightarrow 0$ 。例如,个体 A 为 110010111000, $c=2$,变异基因位置为第 2 位和第 6 位,则个体 A 变异后的新个体为 100011111000。

(2) 实值变异

设 $X=(x_1, x_2, \dots, x_k, \dots, x_l)$ 为群体中的一个个体,若 x_k 为变异点, x_k 的取值范围为 (U_{\min}^k, U_{\max}^k) 。在 x_k 处对个体 $X=(x_1, x_2, \dots, x_k, \dots, x_l)$ 进行变异操作后,可得一个新的个体 $X'=(x_1, x_2, \dots, x'_k, \dots, x_l)$,其中变异点的新基因值为:

$$x'_k = U_{\min}^k + \zeta(U_{\max}^k - U_{\min}^k) \quad (6.3.17)$$

其中, ζ 为 $[0,1]$ 范围内服从均匀概率分布的一个随机数。

6.3.4 遗传算法运行参数的选择

1. 二进制编码串长度 l

对于二进制编码而言,编码串长度 l 的选择与所求解的精度有关,若编码串精度为 δ ,由式 (6.3.1),可得二进制编码串长度 l 为

$$l = \log_2 \left(\frac{U_{\max} - U_{\min} + \delta}{\delta} \right) \quad (6.3.18)$$

2. 群体的大小 M

作为遗传算法控制的参数之一,群体的大小对于遗传算法的效能发挥有一定影响。在我们的认识中,群体的大小 M 应该较大,这样,遗传算法处理的模式较多,生成有意义的基因块并逐渐进化为最优解的机会就越高。也就是说,群体规模越大,群体中个体的多样性越高,算法陷入局部解的危险就越小。但是,群体规模太大,算法的计算量也增大。

对于具体的优化问题,没有理论结果告诉我们如何选择群体的大小。根据有关文献报道,

群体大小的取值范围一般为 $M=10\sim 160$ 。

3. 交叉概率 P_c

交叉概率 P_c 用于控制交叉的频度。较大的交叉概率可增强遗传算法开辟新的搜索区域的能力, 种群中编码串更新很快, 但种群中优良模式的个体会遭受到破坏。若交叉概率取值太小, 交叉产生新个体的速度较慢, 从而会使搜索停滞不前。一般建议交叉概率的取值范围为 $0.4\sim 0.9$ 。

4. 变异概率 P_m

变异概率 P_m 直接影响算法的收敛性和最终解的性能。较大的变异概率会使算法不断地探索新的解空间, 增加模式的多样性, 但较大的变异概率会影响算法的收敛性。较小的变异概率会使变异操作产生新个体的能力和抑制早熟现象的能力变差。一般建议变异概率的取值范围为 $0.001\sim 0.1$ 。

5. 算法终止条件

遗传算法的理论已证明了算法具有以概率 1 收敛的极限性质, 然而, 实际操作不允许它无限地进行下去; 而且, 通常问题的最优解也未必知道, 因此, 需要一种终止算法过程的条件。

一般采用的终止算法过程的条件有:

① 事先给定一个最大的遗传进化代数, 当到达此值时, 就停止运行。一般建议最大的遗传进化代数的取值范围为 $100\sim 1000$ 。

② 连续几代个体平均适应度的差异小于要求的很小的偏差值 ε 。

③ 群体中所有个体适应度的方差小于要求的很小的偏差值 ε 。

6.4 遗传算法应用实例^[18]

考虑如下最优化问题:

$$\max f(x) = x \sin(10 \pi x) + 2.0 \quad x \in [-1, 2] \quad (6.4.1)$$

该目标函数曲线如图 6.4.1 所示。用其他方法不难求得该优化问题的最优解为 $x^* = 1.850548$, $f(x^*) = 3.850274$ 。

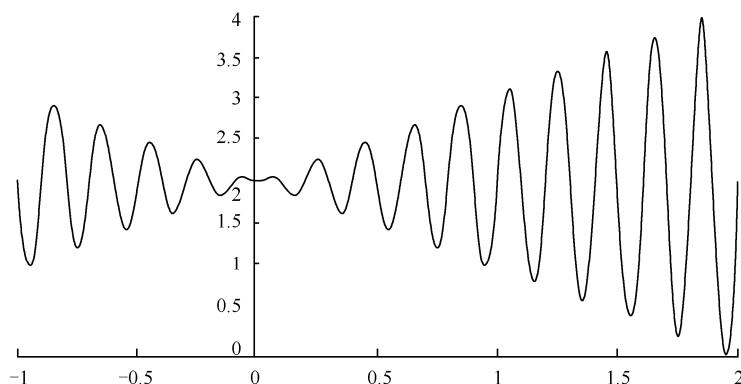


图 6.4.1 目标函数曲线图

下面,我们用遗传算法求解如上最优化问题。

1. 编码

采用二进制编码,变量 x 的取值范围是 $[U_{\min}, U_{\max}] \in [-1, 2]$, 设求解精度确保留 6 位小数, 由式 (6.3.18) 可计算二进制编码串的长度 l 为

$$l \geq \log_2 \left(\frac{U_{\max} - U_{\min} + \delta}{\delta} \right) = \log_2 \left(\frac{2+1+0.000001}{0.000001} \right) = 21.52 \quad (6.4.2)$$

所以,二进制编码的长度 l 取 22 位。

例如,二进制编码串 $S_1 = 1000101110110101000111$, 由式 (6.3.3) 可计算它所对应的实数值为 0.637197。二进制编码串 0000000000000000000000 表示 -1, 二进制编码串 1111111111111111111111 表示 2。

由于优化问题是求极大化, 并且, 在 x 的取值范围 $[-1, 2]$ 内, 目标函数 $f(x)$ 的值大于 0, 因此, 取 $C_{\min} = 0$, 由式 (6.3.7) 可得该优化问题个体的适应度函数 $fit(S)$ 为

$$fit(S) = f(x) = x \sin(10\pi x) + 2.0 \quad (6.4.3)$$

这里,二进制编码串 S 对应变量的值。

例如,有三个个体的二进制编码串为

$$S_1 = 1000101110110101000111$$

$$S_2 = 0000001110000000010000$$

$$S_3 = 1110000000111111000101$$

它们分别对应的变量值为 $x_1 = 0.637197$, $x_2 = -0.958973$, $x_3 = 1.627888$, 个体的适应度函数值分别为

$$fit(S_1) = f(x_1) = 2.586347$$

$$fit(S_2) = f(x_2) = 1.078877$$

$$fit(S_3) = f(x_3) = 3.250654$$

显然,三个个体中 S_3 的适应度函数值最大, S_3 为最佳个体。

2. 遗传操作

(1) 交叉运算

假设经过选择操作(如轮盘赌选择方法)挑选的两个配对父辈个体为 S_2 和 S_3 , 随机选择一个交叉点, 假设交叉点为第 5 位和第 6 位之间的位置, 则交叉运算产生的后代新个体 S'_2 和 S'_3 为

$$\begin{array}{ccc} S_2 \ 00000 \mid 01110000000010000 & \xrightarrow{\text{交叉}} & 0000000000111111000101 \ S'_2 \\ S_3 \ 11100 \mid 00000111111000101 & & 1110001110000000010000 \ S'_3 \end{array}$$

这两个后代新个体 S'_2 和 S'_3 的适应度函数值分别为

$$fit(S'_2) = f(-0.997113) = 1.909673$$

$$fit(S'_3) = f(1.666028) = 3.459252$$

我们注意到个体 S'_3 的适应度函数值比其两个父辈个体的适应度函数值高。

(2) 变异运算

假设已经以一小概率选择了 S_3 的第 5 个遗传因子(即第 5 位)变异, 遗传因子由原来的 0 变为 1, 产生新个体 $\bar{S}_3 = 1110100000111111000101$, 计算个体的适应度函数值 $fit(\bar{S}_3) =$

$f(1.721638)=0.917738$, 发现个体 \bar{S}_3 的适应度函数值比其父辈个体的适应度函数值减少了。但如果 S_3 的第 10 个遗传因子变异, 遗传因子由原来的 0 变为 1, 产生新个体 $\bar{S}'_3=1110000001111111000101$, 计算个体的适应度函数值 $fit(\bar{S}'_3)=f(1.630818)=3.343549$, 又发现个体 \bar{S}'_3 的适应度函数值比其父辈个体的适应度函数值改善了, 这说明了变异操作的“扰动”作用。

3. 遗传算法模拟结果

设定种群大小为 50, 二进制编码串长度为 22, 交叉概率 $P_c=0.25$, 变异概率 $P_m=0.01$, 按照基本遗传算法 (Simple Genetic Algorithm, SGA) 进行运算, 模拟世代的种群中最佳个体的演变情况如表 6.4.1 所示, 运行 89 代时获得的最佳个体为

$$S_{\max}=1111001100111111001011$$

$$fit(S_{\max})=f(1.850549)=3.850274$$

用遗传算法求得的问题 (6.4.1) 的最优解为 $x^*=1.850549$, $f(x^*)=3.850274$ 。这与用其他方法计算出来的最优解非常吻合。

表 6.4.1 模拟世代的种群中最佳个体的演变情况

| 世代数 | 个体的二进制串 | x | 适应度 |
|-----|-------------------------|----------|----------|
| 1 | 1111000110100001110001 | 1.831624 | 3.534806 |
| 4 | 1111001010001101100000 | 1.842416 | 3.790362 |
| 7 | 1111001110011101010111 | 1.854860 | 3.833280 |
| 11 | 1111001110011101010110 | 1.854860 | 3.833286 |
| 17 | 1111001011111101010111 | 1.847536 | 3.842004 |
| 18 | 1111001011111101110000 | 1.847554 | 3.842102 |
| 34 | 11110011011111011000011 | 1.853290 | 3.843402 |
| 40 | 1111001100010001001011 | 1.848443 | 3.846232 |
| 54 | 1111001100010110110001 | 1.848699 | 3.847155 |
| 71 | 1111001101000110110010 | 1.850897 | 3.850162 |
| 89 | 1111001100111111001011 | 1.850549 | 3.850274 |
| 150 | 1111001100111111001011 | 1.850549 | 3.850274 |

注: 在文献[18]中, 表 6.4.1 中个体的二进制串数据前后位置颠倒。

第 7 章 最优化方法在压力加工中的应用

最优化方法在压力加工中的应用实例较多，本书只限于讨论作者在压力加工中所进行的优化设计实例。本章涉及的优化设计实例有：① 多层压配组合挤压凹模优化设计，包括按第四强度理论和疲劳强度理论进行的最优化设计；② 冲裁零件排样优化设计。

7.1 多层压配组合挤压凹模优化设计^[21~23]

因为金属挤压工艺要求挤压成形凹模具有足够的强度，故挤压成形凹模一般采用组合式凹模结构。一至三层挤压成形凹模如图 7.1.1 所示。组合凹模的优化设计就是在挤压工作条件下，优选组合凹模各层零件尺寸参数和配合处的过盈量。使用组合凹模能充分发挥模具材料的潜力、解决凹模开裂、减少凹模尺寸，提高组合凹模的承载能力和提高组合凹模的使用寿命。

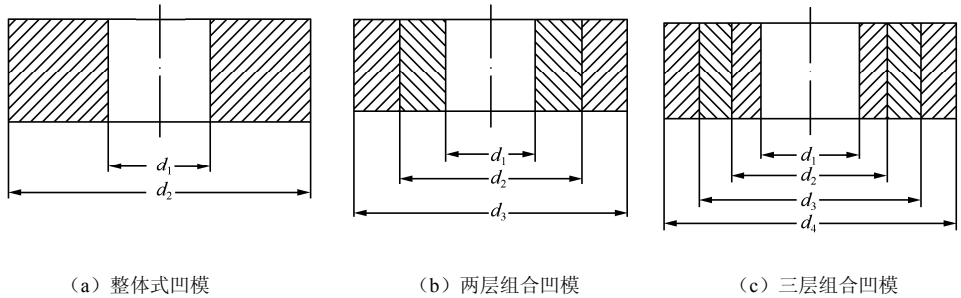


图 7.1.1 挤压成形凹模

7.1.1 组合凹模的特点

多层压配组合挤压凹模的特点是层与层之间采用过盈配合，产生预紧力，挤压时坯料对组合凹模产生工作压力，预紧力和工作压力都会使组合凹模内层产生应力。这两组应力对组合凹模内层来讲，无论是径向应力还是切向应力，它们的作用的方向是相反的。根据力的叠加原理，这两组应力相互抵消，大大改善了组合凹模内层的应力状态，从而提高组合凹模的强度。

组合凹模有如下优点：

(1) 提高了凹模的强度，对一定尺寸的凹模（即凹模的内、外半径相同），二层组合凹模的强度约为整体单层凹模强度的 1.8 倍，三层组合凹模的强度约为整体单层凹模强度的 2 倍。

(其数据见后面组合凹模优化数例);特别是对硬质合金等脆性和抗拉性能差的材料作为挤压成形凹模材料,预应力组合凹模更显其优越性。

(2) 凹模内圈尺寸减小,使耗用的高级合金钢的材料相应减少。原先整个凹模都用高级合金钢制成,而组合凹模仅内圈用高级合金钢材料,中圈与外圈可改用性能较差的其他金属材料。

(3) 由于内圈尺寸减小,热处理变得容易,提高了凹模热处理质量。

(4) 当凹模内圈磨损或损坏后,仅需要调换其内圈,不必整个凹模报废等。

但多层预应力组合凹模也存在加工要求严格、压合工艺要求高等问题。

7.1.2 整体凹模受力分析

一般组合凹模各层均为厚壁圆筒,其结构和受力如图 7.1.2 所示。设 r_1 、 r_2 为厚壁圆筒的内、外半径,在内压 p_1 和外压 p_2 的作用下,假设没有轴向力,即 $\sigma_z=0$ 。由弹性力学可知,截面内任意一点半径 r 处的切向应力 σ_t 与径向应力 σ_r 和径向位移 u 有如下通解^[24]:

$$\begin{cases} \sigma_t = -\frac{A}{r^2} + C \\ \sigma_r = \frac{A}{r^2} + C \\ u = \frac{1}{E} \left[-(1+\nu) \frac{A}{r} + (1-\nu) Cr \right] \end{cases} \quad (7.1.1)$$

式中, ν 和 E 分别为材料的泊桑系数和弹性模量, A 和 C 为待定常数。

式 (7.1.1) 是拉美 (Lame) 提出来的,故称为拉美方程或拉美公式。

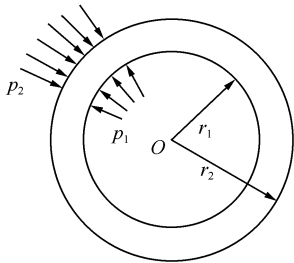


图 7.1.2 厚壁圆筒的受力图

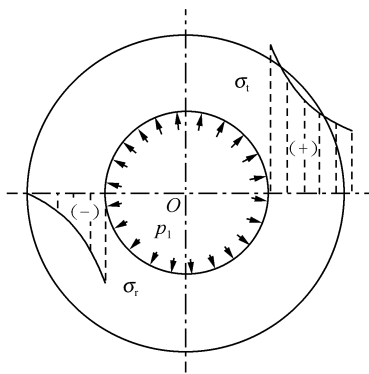


图 7.1.3 厚壁圆筒的应力分布

由边界条件 (即 $r=r_1$, $\sigma_r=p_1$ 和 $r=r_2$, $\sigma_r=-p_2$) 确定 A 和 C , 代入式 (7.1.1) 可得

$$\begin{cases} \sigma_t = \frac{r_1^2 p_1 - r_2^2 p_2}{r_2^2 - r_1^2} + \frac{(p_1 - p_2) r_1^2 r_2^2}{r^2 (r_2^2 - r_1^2)} \\ \sigma_r = \frac{r_1^2 p_1 - r_2^2 p_2}{r_2^2 - r_1^2} - \frac{(p_1 - p_2) r_1^2 r_2^2}{r^2 (r_2^2 - r_1^2)} \\ u = \frac{1-\nu}{E} \cdot \frac{(r_1^2 p_1 - r_2^2 p_2) r}{r_2^2 - r_1^2} + \frac{1+\nu}{E} \cdot \frac{(p_1 - p_2) r_1^2 r_2^2}{r_2^2 - r_1^2} \cdot \frac{1}{r} \end{cases} \quad (7.1.2)$$

当挤压凹模为整体凹模时, 其受力情况与只受内压 p_1 作用的厚壁圆筒完全相同。因此, 将 $p_2=0$ 代入式 (7.1.2), 得整体凹模内任意点 r 处的切向应力 σ_t 和径向应力 σ_r 的计算公式

$$\begin{cases} \sigma_t = \frac{r_1^2 p_1}{r_2^2 - r_1^2} \left(1 + \frac{r_2^2}{r^2} \right) \\ \sigma_r = \frac{r_1^2 p_1}{r_2^2 - r_1^2} \left(1 - \frac{r_2^2}{r^2} \right) \end{cases} \quad (7.1.3)$$

根据式 (7.1.3), 厚壁圆筒的 σ_t 和 σ_r 的分布情况如图 7.1.3 所示。由图可知, σ_t 为拉应力, σ_r 为压应力, 两者的绝对值都是在 $r=r_1$ 处为最大。

下面, 分析整体式凹模存在的问题。按照第四强度理论, 整体式凹模的等效应力 σ_v 由下式计算:

$$\sigma_v = \sqrt{\sigma_t^2 + \sigma_r^2 - \sigma_t \sigma_r} \quad (7.1.4)$$

将式 (7.1.3) 代入上式, 得

$$\sigma_v = \frac{p_1}{k^2 - 1} \sqrt{1 + 3 \frac{r_2^4}{r^4}} \quad (7.1.5)$$

式中, $k = \frac{r_2}{r_1}$ 。

由上式可知, r 越小, σ_v 越大。当 $r=r_1$ 时, σ_v 为最大值, 即内壁处的等效应力最大。由式 (7.1.5) 得

$$\frac{\sigma_{v\max}}{p_1} = \frac{\sqrt{3 + \left(\frac{1}{k}\right)^4}}{1 - \left(\frac{1}{k}\right)^2} \quad (7.1.6)$$

式中, $\sigma_{v\max}$ 为最大等效应力。

根据式 (7.1.6), $\frac{\sigma_{v\max}}{p_1}$ 与 k 的关系曲线如图 7.1.4 所示。

从上述分析可以看出:

(1) 整体式凹模的切向应力 σ_t 为拉应力, 径向应力 σ_r 为压应力, 它们都与作用在凹模内

壁的压力 p_1 成正比。切向拉应力 σ_r 的最大值在凹模内壁，该拉应力足够大时，就能引起凹模从内壁开始的纵向开裂，如图 7.1.5 所示，这是整体凹模存在的最大问题。

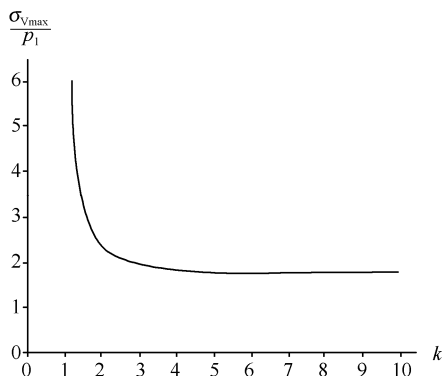


图 7.1.4 σ_{vmax}/p_1 与 k 的关系曲线

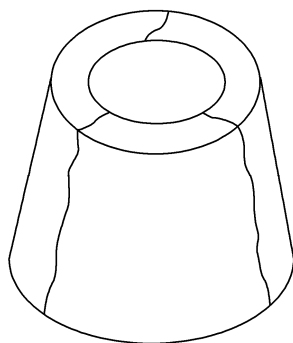


图 7.1.5 整体式凹模的纵向开裂

(2) 凹模内壁处的等效力 σ_v 为最大，即凹模强度的薄弱部位在内壁。因此，验算凹模的强度时，危险断面应取在内壁。

(3) 从图 7.1.4 可知，当 $k \leq 2.5$ 时， $\frac{\sigma_{vmax}}{p_1}$ 有明显地减少；当 $k > 2.5$ 时， $\frac{\sigma_{vmax}}{p_1}$ 减少非常

缓慢。因此，当 $k > 2.5$ 时，增大凹模的壁厚来减少 $\frac{\sigma_{vmax}}{p_1}$ 的方法是不可取的。

7.1.3 组合凹模的力学分析

下面，讨论 3 层组合凹模的力学分析。设 r_i 、 r_{i+1} 、 E_i 、 $\nu_i (i=1,2,3)$ 分别为 3 层组合凹模的第 i 层零件的内半径、外半径、材料的弹性模量和泊桑系数。假设组装时，第 1 层（最内层）和第 2 层先压配合。设两层零件之间的单边过盈量为 Δr_2 ，压配后的预紧力 p_2 。由式 (7.1.2) 可得

$$p_2 = \frac{\Delta r_2}{f_2} \quad (7.1.7)$$

式中， f_2 为柔度系数。

用下式计算 f_2 ：

$$f_2 = \frac{r_2}{E_1} \left(\frac{r_2^2 + r_1^2}{r_2^2 - r_1^2} - \nu_1 \right) + \frac{r_2}{E_2} \left(\frac{r_3^2 + r_2^2}{r_3^2 - r_2^2} + \nu_2 \right) \quad (7.1.8)$$

由式 (7.1.2) 可得，由 p_2 引起的第 1 层和第 2 层零件的应力公式。设 $\Delta \bar{r}_3$ 为由预紧力 p_2 引起中间层外半径 r_3 的扩大量，由式 (7.1.2) 可得

$$\Delta \bar{r}_3 = u(r_3) = \frac{2r_2^2 r_3 p_2}{E_2 (r_3^2 - r_2^2)} \quad (7.1.9)$$

设第2层和第3层零件之间加工预留的单边过盈量为 Δr_3 ，第3层压配时，实际单边过盈量为 $\Delta r_3 + \Delta \bar{r}_3$ 。设压配后的预紧力 p_3 ，设 A_i 和 C_i ($i=1, 2$) 为 $p_3=1$ 时引起组合凹模第 i 层的应力和位移式(7.1.1)中的系数。由边界条件可得

$$\begin{cases} \frac{A_1}{r_1^2} + C_1 = 0 \\ \frac{A_1}{r_2^2} + C_1 = \frac{A_2}{r_2^2} + C_2 \\ \frac{A_2}{r_3^2} + C_2 = -1 \\ \frac{1}{E_1} \left[-(1+\nu_1) \frac{A_1}{r_2} + (1-\nu_1) C_1 r_2 \right] = \frac{1}{E_2} \left[-(1+\nu_2) \frac{A_2}{r_2} + (1-\nu_2) C_2 r_2 \right] \end{cases} \quad (7.1.10)$$

解上式可得 A_i 和 C_i ($i=1, 2$)。

由下式计算预紧力 p_3 ：

$$p_3 = \frac{\Delta r_3 + \Delta \bar{r}_3}{f_3} \quad (7.1.11)$$

式中， f_3 为柔度系数。

用下式计算 f_3 ：

$$f_3 = \frac{r_3}{E_3} \left(\frac{r_4^2 + r_3^2}{r_4^2 - r_3^2} + \nu_3 \right) - \frac{1}{E_2} \left(-(1+\nu_2) \frac{A_2}{r_3} + (1-\nu_2) C_2 r_3 \right) \quad (7.1.12)$$

由式(7.1.10)、式(7.1.11)、式(7.1.12)和式(7.1.1)可得由 p_3 引起的第1层和第2层零件的应力公式，由式(7.1.11)、式(7.1.12)和式(7.1.3)可得由 p_3 引起的第3层零件的应力公式。

另设 \bar{A}_i 、 \bar{C}_i ($i=1, 2, 3$) 为组合凹模第 i 层由工作载荷 p_1 (承载能力) 引起的应力和位移(7.1.1)式中的系数，由边界条件可得

$$\begin{cases} \frac{\bar{A}_1}{r_1^2} + \bar{C}_1 = -p_1 \\ \frac{\bar{A}_1}{r_2^2} + \bar{C}_1 = \frac{\bar{A}_2}{r_2^2} + \bar{C}_2 \\ \frac{\bar{A}_2}{r_3^2} + \bar{C}_2 = \frac{\bar{A}_3}{r_3^2} + \bar{C}_3 \\ \frac{\bar{A}_3}{r_4^2} + \bar{C}_3 = 0 \\ \frac{1}{E_1} \left[-(1+\nu_1) \frac{\bar{A}_1}{r_2} + (1-\nu_1) \bar{C}_1 r_2 \right] = \frac{1}{E_2} \left[-(1+\nu_2) \frac{\bar{A}_2}{r_2} + (1-\nu_2) \bar{C}_2 r_2 \right] \\ \frac{1}{E_2} \left[-(1+\nu_2) \frac{\bar{A}_2}{r_3} + (1-\nu_2) \bar{C}_2 r_3 \right] = \frac{1}{E_3} \left[-(1+\nu_3) \frac{\bar{A}_3}{r_3} + (1-\nu_3) \bar{C}_3 r_3 \right] \end{cases} \quad (7.1.13)$$

解上式可得 \bar{A}_i 、 \bar{C}_i ($i=1, 2, 3$)。将 \bar{A}_i 、 \bar{C}_i 代入式 (7.1.1), 可得由工作载荷 p_1 所引起各层零件的应力。

根据以上计算公式和力的叠加原理, 可求得 3 层组合凹模压配后和在工作状态下各层零件内表面上的应力计算公式 (见表 7.1.1)。

表 7.1.1 3 层组合凹模各层零件内表面上的应力计算公式

| 零件 | 应力 | 内层、中层压配 | 第 3 层压配 | 工作载荷状态 |
|----|------------|--|---|--|
| 1 | σ_r | 0 | 0 | $-p_1$ |
| | σ_t | $-\frac{2r_2^2 p_2}{r_2^2 - r_1^2}$ | $\left(-\frac{A_1}{r_1^2} + C_1\right)p_3 - \frac{2r_2^2 p_2}{r_2^2 - r_1^2}$ | $-\frac{\bar{A}_1}{r_1^2} + \bar{C}_1 + \left(-\frac{A_1}{r_1^2} + C_1\right)p_3 - \frac{2r_2^2 p_2}{r_2^2 - r_1^2}$ |
| 2 | σ_r | $-p_2$ | $\left(\frac{A_2}{r_2^2} + C_2\right)p_3 - p_2$ | $\frac{\bar{A}_2}{r_2^2} + \bar{C}_2 + \left(\frac{A_2}{r_2^2} + C_2\right)p_3 - p_2$ |
| | σ_t | $\frac{(r_2^2 + r_3^2)p_2}{r_3^2 - r_2^2}$ | $\left(-\frac{A_2}{r_2^2} + C_2\right)p_3 + \frac{(r_3^2 + r_2^2)p_2}{r_3^2 - r_2^2}$ | $-\frac{\bar{A}_2}{r_2^2} + \bar{C}_2 + \left(-\frac{A_2}{r_2^2} + C_2\right)p_3 + \frac{(r_3^2 + r_2^2)p_2}{r_3^2 - r_2^2}$ |
| 3 | σ_r | 0 | $-p_3$ | $\frac{\bar{A}_3}{r_3^2} + \bar{C}_3 - p_3$ |
| | σ_t | 0 | $\frac{(r_4^2 + r_3^2)p_3}{r_4^2 - r_3^2}$ | $-\frac{\bar{A}_3}{r_3^2} + \bar{C}_3 + \frac{(r_4^2 + r_3^2)p_3}{r_4^2 - r_3^2}$ |

注: 为了简化, 用 σ_{ij} 和 σ_{ij} 分别代表第 i 行和第 j 列的 σ_r 和 σ_t , $pp_j = -\sigma_{r2j}$ ($i=1, 2, 3; j=1, 2, 3$)。

7.1.4 组合凹模的强度校核

由于组合凹模各层皆为厚壁圆筒, 因此, 组合凹模的强度校核就可以转化为厚壁圆筒的强度校核。下面, 分别按最大剪应力强度理论、畸变能强度理论和疲劳强度理论讨论组合凹模的强度校核。组合凹模的三个主应力一般为: $\sigma_1 = \sigma_t$ 、 $\sigma_2 = \sigma_z = 0$ 和 $\sigma_3 = \sigma_r$ 。

1. 最大剪应力强度理论 (第三强度理论)

$$\tau_{\max} = \frac{\sigma_1 - \sigma_3}{2} = \frac{\sigma_t - \sigma_r}{2} \leq \frac{\sigma_s}{2}$$

即

$$\sigma_t - \sigma_r \leq \sigma_s \quad (7.1.14)$$

式中, σ_s 为材料的屈服极限。

现在, 讨论厚壁筒强度校核点所在的位置, 由式 (7.1.1) 得

$$\sigma_t - \sigma_r = -\frac{2A}{r^2} \quad (7.1.15)$$

由上式可知, 无论内压 p_1 和外压 p_2 为何值, 按最大剪应力强度理论校核, 厚壁筒强度校核点在厚壁筒的内表面上。

2. 畸变能强度理论 (第四强度理论)

当采用畸变能强度条件时, 其等效应力为

$$\sigma_v = \sqrt{\frac{1}{2}[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2]} \leq \sigma_s$$

即

$$\sigma_v = \sqrt{\sigma_r^2 + \sigma_t^2 - \sigma_r \sigma_t} \leq \sigma_s \quad (7.1.16)$$

下面, 讨论厚壁筒强度校核点所在的位置, 由式 (7.1.1) 得

$$\begin{aligned} \sigma_v^2 &= \left(\frac{A}{r^2} + C \right)^2 + \left(-\frac{A}{r^2} + C \right)^2 - \left(\frac{A}{r^2} + C \right) \left(-\frac{A}{r^2} + C \right) \\ &= \frac{3A^2}{r^4} + C^2 \end{aligned} \quad (7.1.17)$$

由式 (7.1.17) 可知, 无论内压 p_1 和外压 p_2 为何值, 厚壁筒的最大等效应力在厚壁筒的内表面上, 因此, 按畸变能强度理论校核, 厚壁筒强度校核点也在厚壁筒的内表面上。

3. 疲劳强度理论

设 σ_{-1} 、 σ_b 、 σ_s 和 σ'_s 分别为材料的疲劳极限、拉伸极限强度、拉伸屈服强度和压缩屈服强度, σ_a 和 σ_m 为单向循环载荷作用下的应力幅和平均应力。单向循环荷载应力疲劳极限应力曲线如图 7.1.6 曲线 ABCD 所示^[25], 其数学式为

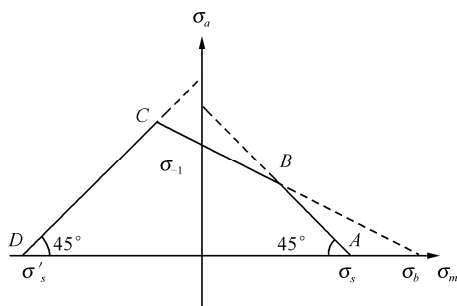


图 7.1.6 单向循环荷载应力疲劳极限曲线

$$\sigma_a = \begin{cases} \sigma'_s + \sigma_m & -\sigma'_s \leq \sigma_m \leq \frac{\sigma_b(\sigma_{-1} - \sigma'_s)}{\sigma_b + \sigma_{-1}} \\ \left(1 - \frac{\sigma_m}{\sigma_b}\right) \sigma_{-1} & \frac{\sigma_b(\sigma_{-1} - \sigma'_s)}{\sigma_b + \sigma_{-1}} \leq \sigma_m \leq \frac{\sigma_b(\sigma_s - \sigma_{-1})}{\sigma_b - \sigma_{-1}} \\ \sigma_s - \sigma_m & \frac{\sigma_b(\sigma_s - \sigma_{-1})}{\sigma_b - \sigma_{-1}} \leq \sigma_m \leq \sigma_s \end{cases} \quad (7.1.18)$$

影响疲劳极限的因素有很多。对于厚壁筒，主要有零件尺寸系数 ε 和零件的表面加工质量系数 β 。零件尺寸系数 ε 为^[26]

$$\varepsilon = (d_0/d)^\alpha \quad (7.1.19)$$

式中， d 和 d_0 分别为零件直径和疲劳实验试件的直径。

对于结构钢，指数 α 在0.08~0.12范围内。按壁厚等效的原则计算厚壁筒的等效直径 d ，厚壁筒的疲劳强度为圆柱形的疲劳极限的0.97倍^[25]。考虑到这些因素，疲劳强度 σ_{-1} 应做适当的调整。设 $\bar{\sigma}_{-1}$ 为调整后的厚壁筒的疲劳强度，则 $\bar{\sigma}_{-1} = 0.97\varepsilon\beta\sigma_{-1}$ 。将 $\bar{\sigma}_{-1}$ 代替式(7.1.18)中的 σ_{-1} ，把相关的强度除以安全系数 n ，就可以进行疲劳强度校核了，即对于挤压组合凹模第 i 层零件，满足 $\sigma_{eai} \leq \sigma_{ai}$ ， σ_{ai} 按式(7.1.18)计算。

用Sines方法^[27]将厚壁筒的循环多向应力等效地转化为单向循环应力。在脉冲循环载荷作用下，组合凹模的第 i 层等效的应力幅 σ_{eai} 和等效平均应力 σ_{emi} 分别为

$$\sigma_{eai} = \sqrt{\sigma_{ari}^2 + \sigma_{ati}^2 - \sigma_{ari}\sigma_{ati}} \quad (7.1.20)$$

$$\sigma_{emi} = \sigma_{mri} + \sigma_{mti} \quad (7.1.21)$$

式中， $\sigma_{a\eta i} = \frac{1}{2}(\sigma_{\eta 3i} - \sigma_{\eta 2i})$ 和 $\sigma_{m\eta i} = \frac{1}{2}(\sigma_{\eta 3i} + \sigma_{\eta 2i})$ ($i=1,2,3$, $\eta=r,t$)， $\sigma_{\eta 2i}$ 等见表7.1.1。

现在，讨论厚壁筒的疲劳强度校核点所在的位置。挤压组合凹模第 i 层零件由压配组装引起的应力为

$$\begin{cases} \sigma_{rli} = A'/r^2 + C' \\ \sigma_{tli} = -A'/r^2 + C' \end{cases} \quad (7.1.22)$$

式中， A' 和 C' 为待定常数。

由工作压力 p_1 引起的应力为

$$\begin{cases} \sigma_{r2i} = A''/r^2 + C'' \\ \sigma_{t2i} = -A''/r^2 + C'' \end{cases} \quad (7.1.23)$$

式中， A'' 和 C'' 为待定常数。

由式(7.1.20)~式(7.1.23)得

$$\begin{cases} \sigma_{emi} = C' + C'' \\ \sigma_{eai}^2 = \frac{3(A'' - A')^2}{4r^4} + \frac{(C'' - C')^2}{4} \end{cases} \quad (7.1.24)$$

式(7.1.24)表明了组合凹模各层的平均应力 σ_{emi} 为常数；在组合凹模各层的内表面上，组合

凹模各层的应力幅 σ_{eai} 最大。因此,挤压组合凹模的各层零件疲劳强度校核点在其内表面上。

从以上讨论知,不管按什么强度理论进行校核,其强度校核点都在组合凹模各层的内表面上。

7.1.5 多层压配组合挤压成形凹模的优化设计

下面,讨论1~3层压配组合挤压凹模的优化设计。

1. 按畸变能强度理论进行优化设计

首先,确定设计变量。可取结构尺寸参数 r_2 、 r_3 和 r_4 ,预紧力 p_2 和 p_3 ,以及组合凹模的承载能力 p_1 (或称工作压力)作为设计变量。

其次,建立目标函数。目标函数可为两类,第一类是给定组合凹模的承载能力,使组合凹模外半径最小,即节省模具材料,从而降低模具费用;第二类是给定凹模的外半径,使凹模的承载能力最大,即提高组合凹模的强度,从而提高模具的使用寿命。

对于1~3层组合凹模,第一类目标函数为

$$\begin{cases} \min r_2(r_2) & \text{对于1层凹模} \\ \min r_3(r_2, r_3, p_2) & \text{对于2层凹模} \\ \min r_4(r_2, r_3, r_4, p_2, p_3) & \text{对于3层凹模} \end{cases} \quad (7.1.25)$$

对于1~3层组合凹模,第二类目标函数为

$$\begin{cases} \max p_1(p_1) & \text{对于1层凹模} \\ \max p_1(r_2, p_1, p_2) & \text{对于2层凹模} \\ \max p_1(r_2, r_3, p_1, p_2, p_3) & \text{对于3层凹模} \end{cases} \quad (7.1.26)$$

最后,建立约束条件。在装配和工作条件下,组合凹模各层内表面上的等效应力不超过材料的许用应力,保证组合凹模各层的壁厚不小于最小壁厚、不大于最大壁厚,组合凹模满足刚度条件,以及预紧力非负的条件。按畸变能强度理论,3层组合凹模的约束条件为

$$\begin{cases} \sqrt{\sigma_{r1i}^2 + \sigma_{t1i}^2 - \sigma_{r1i}\sigma_{t1i}} \leq [\sigma_1] & (i=2,3) \\ \sqrt{\sigma_{r2i}^2 + \sigma_{t2i}^2 - \sigma_{r2i}\sigma_{t2i}} \leq [\sigma_2] & (i=1,2,3) \\ \sqrt{\sigma_{r3i}^2 + \sigma_{t3i}^2 - \sigma_{r3i}\sigma_{t3i}} \leq [\sigma_3] \\ \frac{1}{E_1} \left[-(1+\nu_1) \frac{\bar{A}_1}{r_1} + (1+\nu_1) \bar{C}_1 r_1 \right] < [\delta] \\ r_{i+1} \geq r_i + \Delta r_{i\min} & (i=1,2,3) \\ r_{i+1} \leq r_i + \Delta r_{i\max} & (i=1,2,3) \\ p_2, p_3 \geq 0 \end{cases} \quad (7.1.27)$$

式中, $[\sigma_1]$ 、 $[\sigma_2]$ 和 $[\sigma_3]$ 分别为组合凹模第1层、第2层和第3层材料的许用应力; $[\delta]$ 为组合凹模的刚度条件,即组合凹模内径允许的径向位移; $\Delta r_{i\min}$ 为组合凹模第*i*层的最小壁厚,

$\Delta r_{i\max}$ 为组合凹模第 i 层的最大壁厚, σ_{r1i} 、 σ_{t1i} 、 σ_{r2i} 和 σ_{t2i} 等见表 7.1.1。

删除 3 层组合凹模中多余的约束条件, 并且令 $p_3=0$, 可得 2 层组合凹模的约束条件为

$$\begin{cases} \sqrt{\sigma_{r1i}^2 + \sigma_{t1i}^2 - \sigma_{r1i}\sigma_{t1i}} \leq [\sigma_1] & (i=1,3) \\ \sqrt{\sigma_{r23}^2 + \sigma_{t23}^2 - \sigma_{r23}\sigma_{t23}} \leq [\sigma_2] \\ \frac{1}{E_1} \left[-(1+\nu_1) \frac{\bar{A}_1}{r_1} + (1+\nu_1) \bar{C}_1 r_1 \right] < [\delta] \\ r_{i+1} \geq r_i + \Delta r_{i\min} & (i=1,2) \\ r_{i+1} \leq r_i + \Delta r_{i\max} & (i=1,2) \\ p_2 \geq 0 \end{cases} \quad (7.1.28)$$

删除 3 层组合凹模中多余的约束条件, 并且令 $p_2=0$ 和 $p_3=0$, 可得整体式凹模的约束条件为

$$\begin{cases} \sqrt{\sigma_{r13}^2 + \sigma_{t13}^2 - \sigma_{r13}\sigma_{t13}} \leq [\sigma_1] \\ \frac{1}{E_1} \left[-(1+\nu_1) \frac{\bar{A}_1}{r_1} + (1+\nu_1) \bar{C}_1 r_1 \right] < [\delta] \\ r_2 \geq r_1 + \Delta r_{1\min} \\ r_2 \leq r_1 + \Delta r_{1\max} \end{cases} \quad (7.1.29)$$

例 7.1.1 取 $r_1=70\text{mm}$, $[\sigma_1] = [\sigma_2] = [\sigma_3] = 2000\text{MPa}$, $E_1=E_2=E_3=210000\text{MPa}$, $\nu_1=\nu_2=\nu_3=0.3$, $\Delta r_{1\min}=\Delta r_{2\min}=\Delta r_{3\min}=10\text{mm}$ 。为了便于比较, 取消了最大壁厚和刚度约束。分别用两类目标函数进行优化设计, 采用复合形法求解 (注: 本节其他例也是采用复合形法求解), 求得 1~3 层组合凹模承载能力与最小外半径的关系曲线如图 7.1.7 所示, 1~3 层组合凹模外半径与最大承载能力的关系曲线如图 7.1.8 所示。

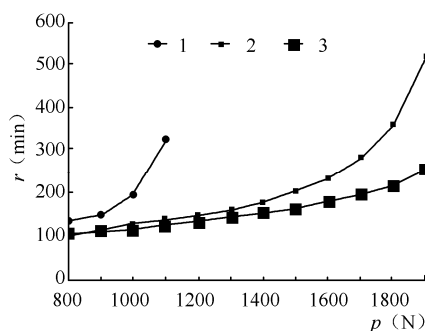


图 7.1.7 1~3 层组合凹模承载能力与凹模最小外半径的关系曲线

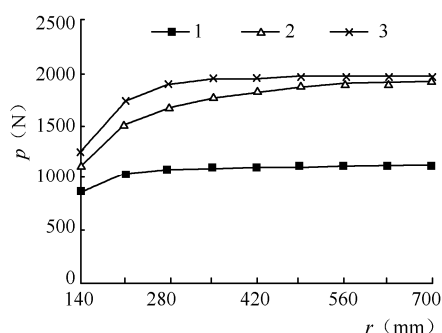


图 7.1.8 1~3 层组合凹模外半径与最大承载能力的关系曲线

例 7.1.2 在例 7.1.1 中, 加入刚度条件 $[\delta] = 0.6\text{mm}$, 用第二类目标函数进行优化设计, 求得 1~3 层组合凹模外半径与最大承载能力的关系曲线如图 7.1.9 所示, 其中 2 层与 3 层组合凹模的最大承载能力完全相同, 这是刚度条件所引起的。

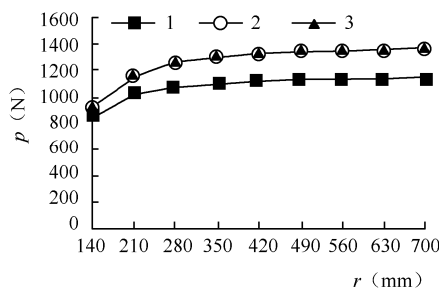


图 7.1.9 1~3 层组合凹模外半径与最大承载能力的关系曲线

例 7.1.3 取 $[\sigma_1] = 2000\text{MPa}$, $[\sigma_2] = 1600\text{MPa}$, $[\sigma_3] = 1100\text{MPa}$, 其余数据与例 7.1.1 相同。当 3 层组合凹模的外半径为 700mm, 用第二类目标函数进行优化, 求得的凹模最大承载能力为 1986.5MPa, 与例 7.1.1 求得的 3 层组合凹模的最大承载能力完全相同。

比较例 7.1.1 与例 7.1.3 可知, 除组合凹模各层使用的材料不同外, 在其他条件完全相同的条件下, 优化后的组合凹模的最大承载能力相同。这说明组合凹模内层使用强度高的材料, 第 2 层使用强度中等的材料, 第 3 层使用强度较低的材料, 这样选择组合凹模的材料是最适合的, 它既能满足组合凹模的强度要求, 又能节省优质模具材料。

2. 按疲劳强度理论进行优化设计

多层压配组合凹模按疲劳强度优化设计的设计变量和目标函数与多层压配组合凹模按畸变能强度理论优化设计的设计变量和目标函数完全相同, 其目标函数参看式 (7.1.25) 和式 (7.1.26)。

最后, 确定约束条件。约束条件有疲劳强度条件、刚度条件, 各层零件最大和最小壁厚, 以及预紧力 p_2 和 p_3 非负条件等。设 $\Delta r_{i\min}$ 、 $\Delta r_{i\max}$ 和 σ_{-1i} 分别为组合凹模第 i 层零件的最小壁厚、最大壁厚和材料的疲劳强度, δ 为组合凹模内半径允许的径向位移。3 层挤压组合凹模的约束条件为

$$\begin{cases} \sigma_{eai} \leq \sigma_{ai} & (i=1, 2, 3) \\ \frac{1}{E_1} \left[-(1+\nu_1) \frac{\bar{A}_1}{r_1} + (1+\nu_1) \bar{C}_1 r_1 \right] < [\delta] \\ r_{i+1} \geq r_i + \Delta r_{i\min} & (i=1, 2, 3) \\ r_{i+1} \leq r_i + \Delta r_{i\max} & (i=1, 2, 3) \\ p_2, p_3 \geq 0 \end{cases} \quad (7.1.30)$$

删除 3 层组合凹模中多余的约束条件, 可得 2 层组合凹模的约束条件为

$$\begin{cases} \sigma_{eai} \leq \sigma_{ai} & (i=1,2) \\ \frac{1}{E_1} \left[-(1+\nu_1) \frac{\bar{A}_1}{r_1} + (1+\nu_1) \bar{C}_1 r_1 \right] < [\delta] \\ r_{i+1} \geq r_i + \Delta r_{i\min} & (i=1,2) \\ r_{i+1} \leq r_i + \Delta r_{i\max} & (i=1,2) \\ p_2 \geq 0 \end{cases} \quad (7.1.31)$$

删除 2 层组合凹模中多余的约束条件, 可得整体式凹模的约束条件为

$$\begin{cases} \sigma_{e a 1} \leq \sigma_{a 1} \\ \frac{1}{E_1} \left[-(1+\nu_1) \frac{\bar{A}_1}{r_1} + (1+\nu_1) \bar{C}_1 r_1 \right] < [\delta] \\ r_2 \geq r_1 + \Delta r_{1\min} \\ r_2 \leq r_1 + \Delta r_{1\max} \end{cases} \quad (7.1.32)$$

例 7.1.4 为了比较起见, 取消了最大壁厚约束条件, 各层零件使用同一种材料。有关参数为: $\sigma_{si}=2000\text{MPa}$, $\sigma'_{si}=2400\text{MPa}$, $\sigma_{bi}=2600\text{MPa}$, $\sigma_{-1i}=1200\text{MPa}$, $E_i=350000\text{MPa}$, $\nu_i=0.3$, $\Delta r_{i\min}=10\text{mm}$, $\beta_i=0.92$, ($i=1, 2, 3$), $\alpha=0.10$, $r_1=70\text{mm}$, $n=1.2$, $\delta=0.4\text{mm}$, 和 $d_0=8\text{mm}$ 。分别对 1~3 层挤压组合凹模的外半径从 140~700mm, 承载能力从 200~1300MPa 一系列数据进行了优化设计, 其优化结果如下:

1~3 层挤压组合凹模的最大承载能力与其外半径的关系曲线如图 7.1.10 所示, 它们的最大承载能力分别为 639MPa、1205MPa 和 1301MPa。从图 7.1.10 可得出结论: ①2 层和 3 层组合凹模的最大承载能力分别为单层凹模的 1.89 和 2.04 倍。②当 1~3 层组合凹模的外半径为内半径的 4 倍时, 它们的承载能力分别为它们最大承载能力的 99.7%、95.3% 和 94.6%。③当外半径大于 350mm, 单层凹模的最大承载能力反而随着外半径的增大而减少, 这是因为凹模材料的疲劳强度值随着外半径的增大而减少的缘故。

1~3 层组合凹模的最小外半径和工作载荷的关系曲线如图 7.1.11 所示。从图 7.1.11 可以看出: ①当工作载荷较小时, 2 层和 3 层组合凹模的外半径反而大于单层凹模的外半径, 这是由于最小壁厚约束条件的缘故。②当承载能力接近于最大的承载能力, 组合凹模的外半径有突然增大的现象。从图 7.1.11 可知, 对于 2 层组合凹模, 当承载能力为 1200MPa 时, 组合凹模的外半径约为 563mm; 当承载能力为 1100MPa 时, 组合凹模的外半径约为 222mm。2 层组合凹模的承载能力仅提高了 $\frac{1200-1100}{1100} \times 100\% \approx 9\%$, 组合凹模的壁厚却提高了

$$\frac{(563-70)-(222-70)}{(222-70)} \times 100\% \approx 224\%。$$

对于 3 层组合凹模, 当承载能力为 1300MPa 时, 组合凹模的外半径约为 673mm; 当承载能力为 1200MPa 时, 组合凹模的外半径约为 245mm。3 层组合凹模的承载能力仅提高约 8%, 组合凹模的壁厚却提高了 245%。因此, 对于 2 层和 3 层组合凹模, 使用组合凹模最大承载能力的 90% 较为合适。

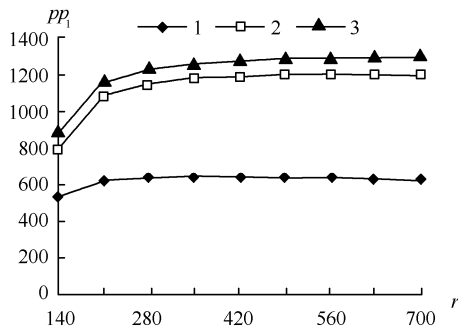


图 7.1.10 1~3 层组合凹模最大承载能力与外半径关系曲线

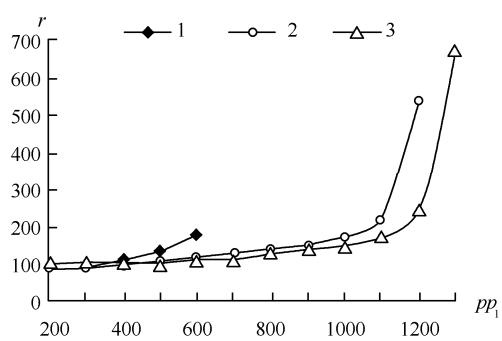


图 7.1.11 1~3 层组合凹模最小外半径与工作载荷关系曲线

承载能力 1200MPa，2 层组合凹模的壁厚是 3 层组合凹模的壁厚的 2.8 倍。因此，合理地确定组合凹模的层数也是非常重要的。

对于 2 层组合凹模，用第一类优化数学模型〔即式 (7.1.25)〕进行优化的结果如表 7.1.2 所示；对于 3 层组合凹模，用第二类优化数学模型〔即式 (7.1.26)〕进行优化的结果如表 7.1.3 所示。

表 7.1.2 2 层组合凹模的优化结果

| | | | | | | | | | | | |
|-------------------|--------|--------|--------|---------|---------|---------|---------|---------|---------|---------|---------|
| pp_1 (MPa) | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000 | 1100 | 1200 |
| pp_2 (MPa) | 137.4 | 144.5 | 209.1 | 267.3 | 332.1 | 403.4 | 481.4 | 566.4 | 673.2 | 774.3 | 844.3 |
| r_2 (mm) | 80.000 | 80.000 | 82.490 | 86.420 | 90.461 | 94.677 | 99.140 | 103.844 | 101.420 | 96.274 | 97.200 |
| R_3 (mm) | 90.000 | 90.000 | 97.553 | 106.667 | 116.776 | 127.925 | 140.134 | 153.388 | 170.720 | 221.563 | 536.476 |
| Δr_2 (mm) | 0.206 | 0.082 | 0.114 | 0.165 | 0.223 | 0.289 | 0.367 | 0.456 | 0.431 | 0.324 | 0.272 |

表 7.1.3 3 层组合凹模的优化结果

| | | | | | | | | | |
|-------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| r_4 (mm) | 140 | 210 | 280 | 350 | 420 | 480 | 560 | 630 | 700 |
| r_2 (mm) | 94.991 | 80.000 | 80.000 | 80.000 | 80.000 | 80.000 | 80.000 | 80.000 | 80.000 |
| r_3 (mm) | 122.040 | 120.062 | 130.940 | 123.389 | 127.403 | 128.380 | 125.681 | 131.212 | 133.360 |
| Δr_2 (mm) | 0.363 | 0.051 | 0.034 | 0.003 | 0.026 | 0.030 | 0.032 | 0.035 | 0.037 |
| Δr_3 (mm) | 0.206 | 0.311 | 0.304 | 0.293 | 0.255 | 0.243 | 0.229 | 0.232 | 0.231 |
| pp_1 (MPa) | 893.4 | 1165.8 | 1231.3 | 1261.3 | 1277.6 | 1287.3 | 1293.6 | 1298.0 | 1301.1 |
| pp_2 (MPa) | 617.3 | 976.5 | 1034.1 | 1060.5 | 1074.9 | 1083.5 | 1089.0 | 1092.9 | 1095.6 |
| pp_3 (MPa) | 224.2 | 571.1 | 583.1 | 676.7 | 638.7 | 636.5 | 649.8 | 623.3 | 614.5 |

通过上述讨论和多层压配组合挤压凹模按疲劳强度优化实例的计算，可得到关于多层压配组合挤压凹模设计的一些结论：

(1) 组合凹模的强度校核点在各层零件的内表面上。

(2) 由于 2 层和 3 层组合凹模的最大承载能力分别为单层凹模的 1.89 和 2.04 倍, 因此, 没有必要使用 3 层以上的组合凹模。

(3) 组合凹模的外半径没有必要大于其内半径的 4 倍, 事实上, 1~3 层组合凹模的外半径为内半径的 4 倍时, 它们的承载能力分别为它们最大承载能力的 99.7%、95.3% 和 94.6%。

(4) 当承载能力接近于最大的承载能力, 1~3 层组合凹模的外半径都有突然增大的现象。使用组合凹模最大承载能力的 90% 较为合适。

7.2 冲裁零件排样优化设计^[28]

7.2.1 冲裁零件排样

冲裁工艺是通过冲头与凹模, 利用冲压设备加压力于冲头和凹模间的被冲材料, 实现冲裁零件从被冲材料分离的工序, 如图 7.2.1 所示。

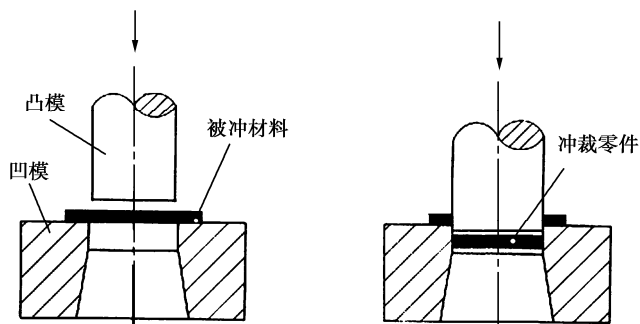
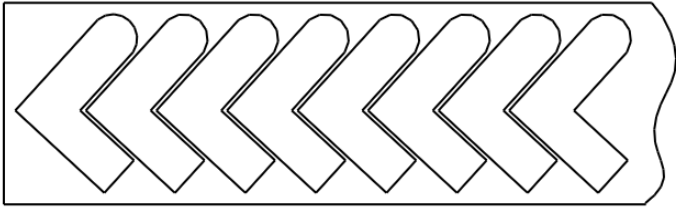

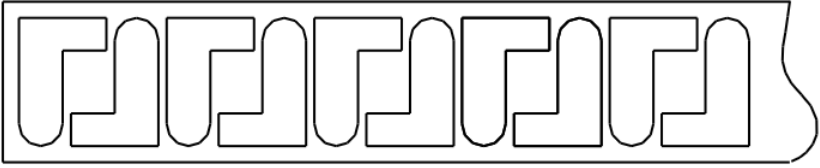
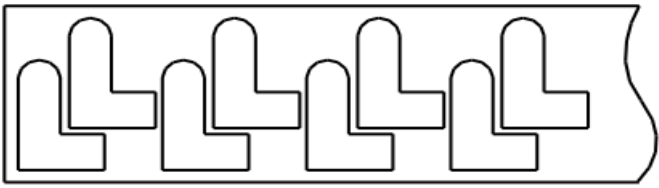
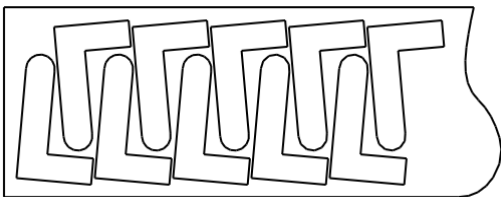
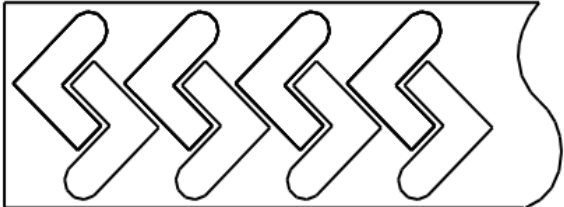


图 7.2.1 冲裁示意图

冲裁零件排样是指冲裁件在板料、条料或带料上的布置方式。常见的冲材零件的排样有: 根据零件排样的排数分为单排和双排; 根据排样方式分为普通、翻转和兑头排样。冲裁零件常见的排样方式如表 7.2.1 所示^[29]。

冲裁零件排样合理与否, 对材料利用率的高低有着直接的影响, 从而影响冲裁零件的成本。在冲裁零件成本中, 材料费用占整个冲裁零件费用的 60%~80%。提高材料利用率是降低冲裁零件成本的主要途径, 而材料的利用率决定于冲裁零件的排样。对于轮廓曲线较为复杂的冲裁零件, 仅凭人的经验进行冲裁零件排样几乎不可能达到最高的材料利用率。

表 7.2.1 常见的冲裁零件排样方式

| 排样方式 | 排样图示 |
|------|---|
| 单排 |  A diagram showing a single row of L-shaped parts nested together. The parts are arranged in a zig-zag pattern, with the long leg of one part fitting into the notch of the next. The entire row is enclosed in a rectangular frame with a wavy right edge. |
| 翻转单排 |  A diagram showing a single row of L-shaped parts nested together, with alternating orientation. The parts are arranged in a zig-zag pattern, with the long leg of one part fitting into the notch of the next. The entire row is enclosed in a rectangular frame with a wavy right edge. |
| 兑头单排 |  A diagram showing a single row of L-shaped parts nested together, with interlocking heads. The parts are arranged in a zig-zag pattern, with the long leg of one part fitting into the notch of the next. The entire row is enclosed in a rectangular frame with a wavy right edge. |
| 普通双排 |  A diagram showing a double row of L-shaped parts nested together. The parts are arranged in two rows, with the long leg of one part fitting into the notch of the next. The entire row is enclosed in a rectangular frame with a wavy right edge. |
| 翻转双排 |  A diagram showing a double row of L-shaped parts nested together, with alternating orientation. The parts are arranged in two rows, with the long leg of one part fitting into the notch of the next. The entire row is enclosed in a rectangular frame with a wavy right edge. |
| 兑头双排 |  A diagram showing a double row of L-shaped parts nested together, with interlocking heads. The parts are arranged in two rows, with the long leg of one part fitting into the notch of the next. The entire row is enclosed in a rectangular frame with a wavy right edge. |

7.2.2 冲裁零件排样优化目标函数的建立

材料利用率 η 是冲裁零件的实际面积 F_0 与原材料面积 F 的百分比,即

$$\eta = \frac{F_0}{F} \times 100\% \quad (7.2.1)$$

冲裁零件排样优化是以实现材料利用率最高作为目标函数,即

$$\max \eta = \max \left\{ \frac{F_0}{F} \times 100\% \right\} \quad (7.2.2)$$

为了便于计算,作者将冲裁零件外轮廓图形加二分之一搭边值后的轮廓图形称为排样零件。在普通、翻转和兑头单、双排冲裁零件排样方式中,普通单排排样方式,排样单元内只含一个冲裁零件;其余冲裁零件排样方式,排样单元内含两个冲裁零件。为了统一起见,普通单排排样方式亦用两个零件作为一个排样单元。

为了简化计算,使用材料利用率等价的参数作为冲裁零件排样优化的目标函数。下面,在原材料分别为带料和板料的情况下,讨论材料利用率等价的参数。

1. 带料

将带料视为“无限长”的条料。所谓“无限长”条料是指条料的长度远远大于排样单元的步长,在计算材料的利用率时,可以不考虑第一个排样单元多损耗的长度及端部工艺要求的损耗的长度。第一个排样单元多损耗的长度为第一个排样单元所耗原材料的长度减去非第一个排样单元所耗原材料的长度。因此,冲裁零件材料利用率按下式计算:

$$\eta = \frac{A_0}{A} \times 100\% \quad (7.2.3)$$

式中, A_0 为一个排样单元内冲裁零件的面积, A 为一个排样单元所消耗的原材料面积。

A 按下式进行计算:

$$A = WH \quad (7.2.4)$$

式中, W 为排样单元的步长, H 为带料的宽度。

由于 A_0 为常数,由式(7.2.3)可知,使材料利用率最高等价于使一个排样单元所消耗的原材料面积 A 最小。因此,对于原材料为带条的情况,冲裁零件排样优化的目标函数为

$$\min A = \min \{WH\} \quad (7.2.5)$$

2. 板料

用板料作为冲裁零件的原材料,是用剪板机将板料剪成一定宽度的条料,再在条料上冲裁零件。对于板料,冲裁零件的材料利用率为

$$\eta = \frac{nS}{LB} \times 100\% \quad (7.2.6)$$

式中, n 为一张板料上冲裁零件的个数, S 为一个冲裁零件的面积, L 和 B 分别为板料的长度和宽度。

由于 S 、 L 和 B 为常数,由式(7.2.6)可知,使材料利用率最高等价于使一张板料上冲裁

零件个数最多, 因此, 对于原材料为板料的情况, 冲裁零件排样优化的目标函数为

$$\max \{n\} \quad (7.2.7)$$

每块板料冲裁零件的个数等于每块板料剪切的条数乘以每一条条料冲裁零件的个数。下面讨论 n 的计算。

(1) 按板料长度方向下料

所谓按长度方向下料是指从板料上用剪板机剪切下来的条料的长度等于板料的长度。

若每一条条料冲裁偶数个零件, 则 n 为

$$n=2 [B/H] [1+(L-W_1)/W] \quad (7.2.8)$$

若每一条条料冲裁奇数个零件, 则 n 为

$$n= [B/H] [1+2(L-W_2)/W] \quad (7.2.9)$$

式中, W 为排样单元的步长, H 为条料的宽度, W_1 为第一排样单元所占条料的长度与两端工艺要求所需损耗的长度之和, W_2 为第一个零件所占条料的长度与两端工艺要求所需损耗的长度之和, $[\]$ 表示取整。

(2) 按板料宽度方向下料

所谓按宽度方向下料是指从板料上用剪板机剪切下来的条料的长度等于板料的宽度。

若每条条料冲裁偶数个零件, 则 n 为

$$n=2 [L/H] [1+(B-W_1)/W] \quad (7.2.10)$$

若每条条料冲裁奇数个零件, 则 n 为

$$n= [L/H] [1+2(B-W_2)/W] \quad (7.2.11)$$

7.2.3 条料宽度和排样单元步长的计算

在冲裁零件排样优化中, 设计变量分别是零件的旋转角度 α 和第二个零件相对于第一个零件在 y 轴上移动的距离 h , 如图 7.2.2 所示。

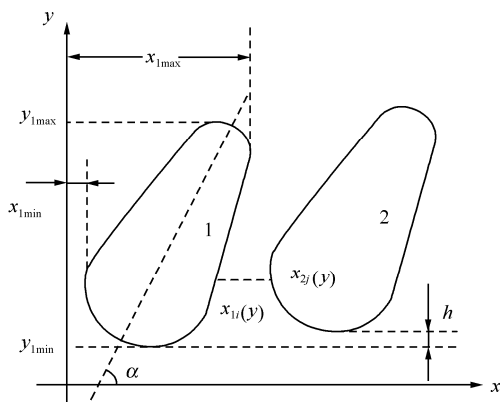


图 7.2.2 排样有关参数示意图

1. 条料宽度 H 的计算

条料宽度是零件的旋转角度 α 和第二个零件相对于第一个零件在 y 轴上移动的距离 h 的函

数, 条料宽度 H 可按下式计算:

$$H(\alpha, h) = H_1(\alpha) + h + \delta h \quad (7.2.12)$$

式中, δh 为一个搭边量与板料每下一条条料剪切的损耗量之和, $H_1(\alpha)$ 为第一个排样零件的高度, 即

$$H_1(\alpha) = y_{1\max} - y_{1\min} \quad (7.2.13)$$

式中, $y_{1\max}$ 和 $y_{1\min}$ 分别为第一个排样零件轮廓线的最大和最小的 y 坐标值, 如图 7.2.2 所示。

2. 排样单元步长 W 的计算

首先, 按设计变量 α 、 h 和排样方式的关系将第二个零件在 y 轴上的相对位置摆正确, 而它们在 x 轴上的位置可以任意, 即不要求它们相互相切, 如图 7.2.2 所示。

其次, 定义两个排样零件轮廓线之间的水平距离函数为

$$d(y) = x_{2l}(y) - x_{1l}(y) \quad (7.2.14)$$

式中, $x_{k,l}(y)$ ($k=1,2;l=i,j$) 为第 k 个排样零件第 l 条线段上的 x 坐标值。

令

$$d_{\max} = \max d(y) \quad (7.2.15)$$

$$d_{\min} = \min d(y) \quad (7.2.16)$$

式中, d_{\min} 的几何意义是两个排样零件相互相切, 且第二个零件在第一个零件的右边时, 它们相对 x 轴上的平移量。

类似于式 (7.2.14) 定义第一排样零件轮廓线之间的水平距离函数

$$d_1(y) = x_{1j}(y) - x_{1i}(y) \quad (7.2.17)$$

令

$$d_{1\max} = \max d_1(y) \quad (7.2.18)$$

排样单元的步长是两个排样零件相切时, 它们的轮廓线之间的最大水平距离, 故排样单元的步长为

$$W(\alpha, h) = \max \{d_{\max} - d_{\min}, d_{1\max}\} \quad (7.2.19)$$

由上面的讨论可知, 排样单元的步长可简单地通过计算轮廓线之间的最大和最小水平距离来确定。下面, 讨论最大和最小水平距离的计算。由于冲裁件的轮廓线一般由直线和圆弧组成, 因此, 仅讨论直线段和圆弧段之间的最大和最小水平距离的计算。

(1) 直线段与直线段

由于直线段与直线段之间的水平距离函数 $d(y)$ 和 $d_1(y)$ 为线性函数, 而线性函数在端点处达到最大或最小, 因此只要比较直线段端点处的水平距离函数值就能确定直线段和直线段之间的最大和最小水平距离。

(2) 直线段与圆弧段

直线段与圆弧段之间的最大和最小水平距离如图 7.2.3 所示, 设直线段与 x 轴的夹角为 α_1 , 则 1 点和 2 点分别为直线段与圆弧段之间的水平距离函数的极值点, 它们相应的圆心角为

$$\beta_1 = 90^\circ + \alpha_1 \quad (7.2.20)$$

$$\beta_2 = 270^\circ + \alpha_1 \quad (7.2.21)$$

式中, α_1 为直线段与 x 轴的夹角。

比较驻点处 (即 β_1 和 β_2 处)、直线段的端点和圆弧段的端点处的水平距离就能确定直线段

与圆弧段之间水平距离的最大值和最小值。

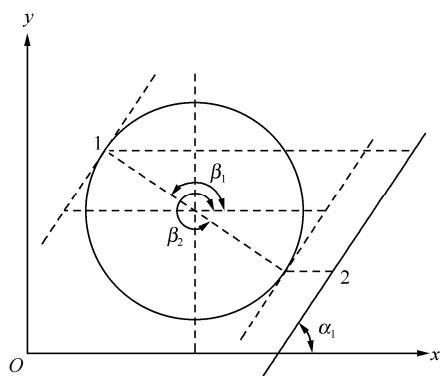


图 7.2.3 直线段与圆弧段水平距离极值点示意图

(3) 圆弧段与圆弧段

设 x_i 、 y_i 、 r_i ($i=1, 2$) 分别为第 i 个排样零件某圆弧段的圆心坐标和半径, 由式 (7.2.14) 得

$$d(y) = x_2 - x_1 \pm \sqrt{r_1^2 - (y - y_1)^2} \pm \sqrt{r_2^2 - (y - y_2)^2} \quad (7.2.22)$$

为了确定圆弧段与圆弧段之间的最大和最小水平距离, 求水平距离函数 $d(y)$ 的驻点。由 $d'(y) = 0$, 得水平距离函数 $d(y)$ 的驻点为:

当 $r_1 = r_2$ 时, 有

$$y^* = \frac{y_1 + y_2}{2} \quad (7.2.23)$$

当 $r_1 \neq r_2$ 时, 有

$$\begin{cases} y_1^* = \frac{r_1 y_2 - r_2 y_1}{r_1 - r_2} \\ y_2^* = \frac{r_1 y_2 + r_2 y_1}{r_1 + r_2} \end{cases} \quad (7.2.24)$$

比较驻点 y_1^* 与 y_2^* (或 y^*) 和两圆弧段的端点处的水平距离函数来确定圆弧段与圆弧段之间的最大和最小水平距离。

3. W_1 和 W_2 的计算

第一个排样单元和第一个排样零件所占条料长度分别用如下两个公式计算:

$$W_1(\alpha, h) = x_{2\max} - x_{1\min} - d_{\min} + \delta b \quad (7.2.25)$$

$$W_2(\alpha, h) = x_{1\max} - x_{1\min} + \delta b \quad (7.2.26)$$

式中, $x_{i\max}$ 和 $x_{i\min}$ ($i=1, 2$) 分别为第 i 个排样零件轮廓线的最大和最小的 x 坐标值, 如图 7.2.2 所示, δb 为一个搭边量与条料两端工艺要求的损耗量之和。

由于排样优化设计目标函数的离散性, 并且优化设计变量只有 2 个, 适宜采用优化方法中的枚举法。所谓枚举法就是枚举出可行解集合内的所有可行解, 从中求出最优解。对于连续函数, 该方法要求对其进行离散化处理。对于设计变量数多的优化设计问题, 枚举法的效率比较低, 有时甚至在目前先进的计算机上也难以求解。用枚举法求解排样优化设计问题的具体做法是: 将总的旋转角度 α 等分为 n 份, 将第二个零件相对于第一个零件在 y 轴上总的移动距离 h 等分为 m 份, 用枚举法进行求解, 假设求得的最优点为 $(\alpha_i, h_j)^T$ 。再将总的旋转角度缩小为 $\alpha = \alpha_{i+1} - \alpha_{i-1}$, 总的移动距离缩小为 $h = h_{j+1} - h_{j-1}$, 再等分再求解, 如此反复计算, 最终求得最优解。

7.2.4 隔离开关 GN2-10/2000 的支承条零件排样优化设计

选用某厂生产的隔离开关 GN2-10/2000 的支承条零件作为冲裁零件排样优化设计的实例。原材料为 2000×1000 的钢板规格, 厚度为 5mm。 $\delta b = 5\text{mm}$, $\delta h = 5\text{mm}$, 搭边值为 5mm。

分别用普通、翻转和兑头排样方式对该零件进行排样优化设计, 其各自的优化结果分别如表 7.2.2 和图 7.2.4 所示。

表 7.2.2 支承条零件排样优化结果

| 排样方式 | α^* | h^* | n | W | H | W_1 | W_2 | 下料方式 |
|------|------------|-------|-----|--------|--------|--------|-------|-------|
| 普通 | 70° | 0 | 272 | 115.38 | 124.45 | 148.41 | 90.72 | 按长度方向 |
| 翻转 | 90° | 32 | 252 | 94.08 | 161.00 | 108.04 | 61.00 | 按长度方向 |
| 兑头 | 90° | 0 | 285 | 102.58 | 129.00 | 106.97 | 61.00 | 按宽度方向 |

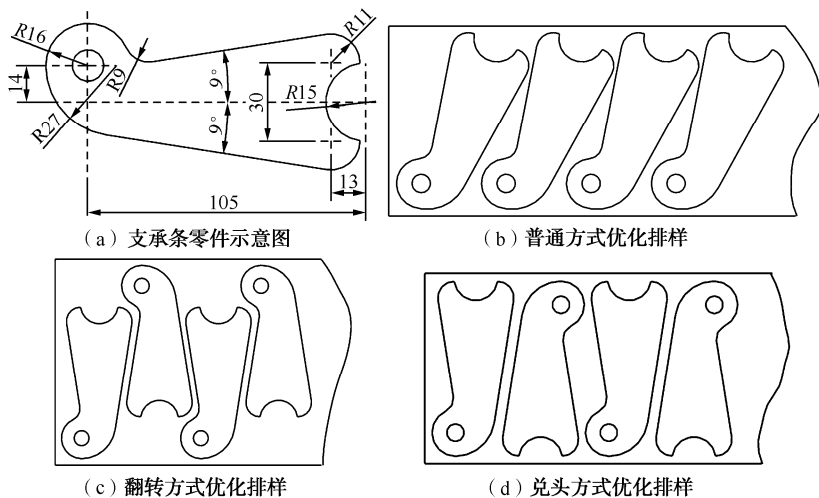


图 7.2.4 支承条零件排样优化设计

通过优化设计可知, 隔离开关 GN2-10/2000 的支承条零件以兑头排样方式和按板料宽度方向下料, 能最大限度地提高材料的利用率。

第 8 章 最优化方法在机构设计中的应用

最优化方法在机构设计中的应用实例较多，本书只限于讨论作者在机构设计中所进行的优化设计实例。在这些实例中，有些是作者首次独立完成的；有些是在其他学者研究的基础上，用改进目标函数等方法进行的优化设计的完善工作。本章涉及的机构优化设计实例有：①飞剪机剪切机构的优化设计；②蟹爪式装载机扒取机构的优化设计；③热锯机送进机构的优化设计；④J88 型冷挤压机拉力肘杆机构优化设计；⑤JA88 型冷挤压机压力肘杆机构优化设计；⑥八连杆双动拉延压力机内外滑块执行机构优化设计；⑦双动拉延压力机内滑块十杆机构的优化设计。

8.1 飞剪机剪切机构的优化设计^[30]

飞剪机是连续轧钢机组中重要而复杂的辅助设备。它的任务是将运行着的轧制钢坯按照工艺要求的尺寸横向剪断。

有各种类型的飞剪机，按照飞剪机剪切机构的种类来分有：滚筒式飞剪机、曲柄连杆式飞剪机、曲柄偏心式飞剪机、摆式飞剪机和曲柄摇杆式飞剪机等。

曲柄连杆式飞剪机的结构如图 8.1.1 所示^[31]。

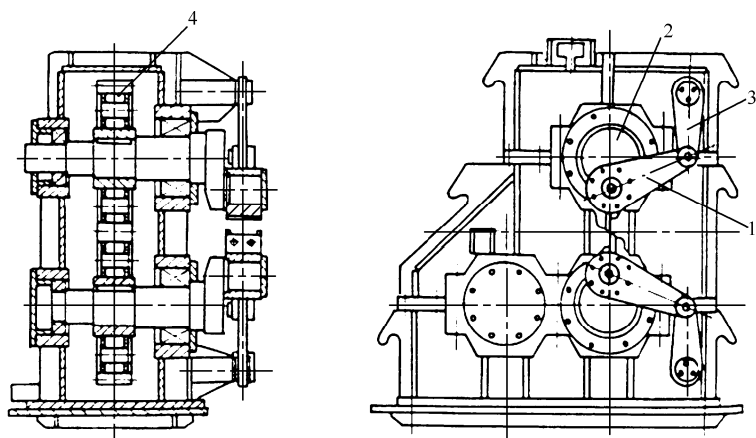


图 8.1.1 曲柄连杆式飞剪机
1—刀架 2—曲柄轴 3—摆杆 4—传动齿轮

8.1.1 对飞剪机剪切机构的要求

飞剪机的任务是将运行着的轧制钢坯按照工艺要求的尺寸横向剪断。由于轧制钢坯的运动,要求飞剪机的两个刀刃不仅要有剪切动作(两个剪切刀刃垂直运动),而且还要有水平方向上的运动。飞剪机剪切机构的结构如图 8.1.2 所示。根据飞剪机的功能作用,对剪切机构有如下要求^[32]:

(1) 为了保证剪断轧制钢坯,两个剪切刀刃运动轨迹曲线具有一定的重叠度 ξ ,如图 8.1.2 (a) 所示。

(2) 在剪切过程中,为了减少两个剪切刀刃的阻力和防止刀刃相互干涉,在剪切区域内两个剪切刀刃垂直于轧制钢坯,如图 8.1.2 (b) 所示。

(3) 在剪切过程中,要求剪切刀刃在水平方向上的速度 V_x 尽量均匀,并且等于或大于 2%~3% 轧制钢坯的运行速度 V_0 , 即 $V_x = (1 \sim 1.03)V_0$ 。

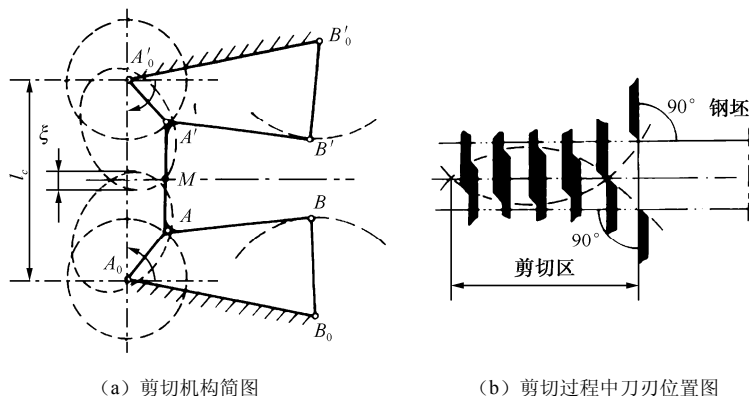


图 8.1.2 飞剪机的剪切机构

8.1.2 飞剪机构的运动分析

飞剪机剪切机构计算简图如图 8.1.3 所示,图中的符号意义分别为:

l_c —— 对称四连杆机构的中心距;

ξ —— 两个剪切刀刃运动轨迹曲线的重叠度;

α —— $x'A_0y'$ 和 xA_0y 坐标系的相对位置角,逆时针方向为正(图 8.1.2 和图 8.1.3 中 α 为负值);

θ —— 连杆 MA 与连杆 AB 的夹角,以连杆 AB 起计逆时针方向为正;

ϕ —— 曲柄 A_0A 与 x' 轴的夹角;

δ —— 连杆 AB 与 x' 轴的夹角;

l_i —— 第 i 构件的长度, $i=1,2,3,4,5$ 。

$$a = \frac{l^2 + l_3^2 - l_2^2}{2l_1 l_3} \quad (8.1.12)$$

$$b = \frac{l_4 - x'_A}{l_1} \quad (8.1.13)$$

$$z = \frac{l^2}{l_1^2} = b^2 + \sin^2 \phi \quad (8.1.14)$$

$$w = \frac{l \sin \psi_2}{l_1} = \sqrt{z - a^2} \quad (8.1.15)$$

连杆上 M 点在 $x'_A y'_A$ 坐标系中的坐标值为

$$x'_M = x'_A + l_5 (\cos \theta \cos \delta - \sin \theta \sin \delta) \quad (8.1.16)$$

$$y'_M = y'_A + l_5 (\sin \theta \cos \delta + \cos \theta \sin \delta) \quad (8.1.17)$$

式中

$$\cos \delta = \frac{x'_B - x'_A}{l_2} \quad (8.1.18)$$

$$\sin \delta = \frac{y'_B - y'_A}{l_2} \quad (8.1.19)$$

利用坐标系的旋转变换关系式, M 点在坐标系 $x_A y_A$ 中的坐标值为

$$x_M = x'_M \cos \alpha - y'_M \sin \alpha \quad (8.1.20)$$

$$y_M = x'_M \sin \alpha + y'_M \cos \alpha \quad (8.1.21)$$

将式 (8.1.20) 对时间求导, 得 M 点水平方向上的速度为

$$V_x = V'_x \cos \alpha - V'_y \sin \alpha \quad (8.1.22)$$

这里

$$V'_x = V'_{Ax} + \frac{l_5 [\cos \theta (V'_{Bx} - V'_{Ax}) - \sin \theta (V'_{By} - V'_{Ay})]}{l_2} \quad (8.1.23)$$

$$V'_y = V'_{Ay} + \frac{l_5 [\sin \theta (V'_{Bx} - V'_{Ax}) + \cos \theta (V'_{By} - V'_{Ay})]}{l_2} \quad (8.1.24)$$

式中

$$V'_{Ax} = -l_1 \omega \sin \phi \quad (8.1.25)$$

$$V'_{Ay} = l_1 \omega \cos \phi \quad (8.1.26)$$

$$V'_{Bx} = l_3 \psi' \sin \psi \quad (8.1.27)$$

$$V'_{By} = l_3 \psi' \cos \psi \quad (8.1.28)$$

而

$$\psi' = \frac{(l_4^2 - l_1^2 - l^2) l_1 \omega \sin \phi}{2l^3 \sqrt{1 - b^2/z}} + \frac{(l_3^2 - l_2^2 - l^2) l_1 l_4 \omega \sin \phi}{2l_3 l^3 \sqrt{1 - a^2/z}} \quad (8.1.29)$$

式中, ω 为曲柄的角速度。

连杆 MA 与 x 轴的位置角为

$$\bar{\delta} = \theta + \delta + \alpha \quad (8.1.30)$$

式中

$$\delta = \arcsin \frac{y'_B - y'_A}{l_2} \quad (8.1.31)$$

8.1.3 剪切机构的优化设计数学模型的建立

1. 设计变量

以飞剪机剪切机构的参数 l_1 、 l_2 、 l_3 、 l_4 、 l_5 、 α 和 θ 作为设计变量。为了使剪切机构的设计更加合理和减少设计变量个数和计算量, 对该机构附加的两个条件是: ①当曲柄与 y 轴重合时, 曲柄与连杆 AM 在同一条铅垂线上; ②两个剪切刀刃的运动轨迹曲线的重叠度 ξ 精确地为某一特定值, 如图 8.1.4 所示。

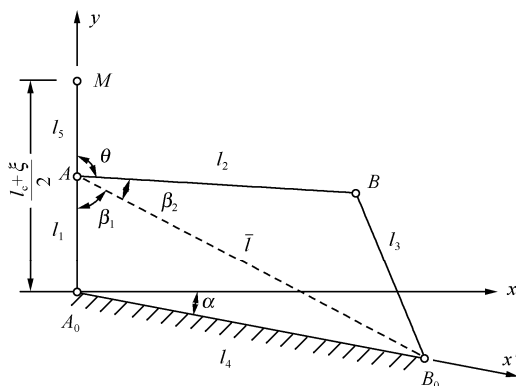


图 8.1.4 飞剪机构附加条件简图

由于附加了以上两个条件, 使 l_1 、 l_2 、 l_3 、 l_4 、 l_5 、 α 和 θ 变量中 l_5 和 θ 变为非独立变量, 由几何关系得

$$l_5 = \frac{l_c + \xi}{2} - l_1 \quad (8.1.32)$$

$$\theta = 180^\circ - \beta_1 - \beta_2 \quad (8.1.33)$$

式中

$$\beta_1 = \arcsin\left(\frac{l_4 \cos \alpha}{\bar{l}}\right) \quad (8.1.34)$$

$$\beta_2 = \arccos\left(\frac{\bar{l}^2 + l_2^2 - l_3^2}{2\bar{l}l_2}\right) \quad (8.1.35)$$

这里

$$\bar{l} = \sqrt{l_1^2 + l_4^2 + 2l_1l_4 \sin \alpha} \quad (8.1.36)$$

剪切机构的设计变量 X 由该机构的独立参数确定, 即设计变量 X 为

$$X = (x_1, x_2, x_3, x_4, x_5)^T = (l_1, l_2, l_3, l_4, \alpha)^T \quad (8.1.37)$$

2. 目标函数

根据上面所述的对剪切机构的 3 条要求。第 (1) 条要求为了保证剪断轧制钢坯, 两个剪刀刃运动轨迹曲线具有一定的重叠度 ξ 。由于上面的附加条件②, 使第 (1) 条要求自动满足。

第 (2) 条要求是在剪切过程中, 要求剪刀刀刃在水平方向上的速度 V_x 尽量均匀, 并且等于或略大于轧制钢坯的运行速度。在剪切工作区间, 应满足剪刀刀刃 M 点在 x 轴方向上的速度 V_x 尽量与轧件的速度相等。我们可用如下函数来衡量它们之间的误差:

$$f_1(X) = \sum_{i=2}^k |V_x(X, \phi_i) - V_{x1}| \quad (8.1.38)$$

式中, k 为剪切工作区曲柄所在的角区间的分割点数, ϕ_i 为第 i 个分割点曲柄与 x' 轴的夹角, V_{x1} 为剪刀刀刃开始剪切时要求 M 点水平方向上的速度。为了防止剪刀刀刃的损坏和考虑到该机构在剪切过程中 V_x 逐渐增加等因素, V_{x1} 可取等于或略小于轧件的运动速度。

第 (3) 条要求是在剪切过程中, 为了减少两个剪刀刀刃的阻力和防止刀刃相互干涉, 在剪切区域内两个剪刀刀刃垂直于轧制钢坯。连杆 MA 与 x 轴的位置角尽量等于 90° , 因此, 我们可用如下函数来衡量在剪切工作区内, 连杆 MA 与 x 轴的位置角与 90° 之间的误差:

$$f_2(X) = \sum_{i=1}^k |\bar{\delta}(X, \phi_i) - 90^\circ| \quad (8.1.39)$$

从上面对目标函数的讨论可知, 飞剪机剪切机构的优化设计问题是一个多目标函数的优化设计问题, 我们用加权方法将多目标函数转化为单目标函数, 即

$$f(X) = W_1 f_1(X) + W_2 f_2(X) \quad (8.1.40)$$

式中, W_1 和 W_2 为加权因子。

3. 约束条件

约束条件由曲柄连杆机构存在、最小传动角的要求和飞剪机结构布置等确定。

曲柄连杆机构存在的条件为

$$\begin{aligned} x_2 - x_1 &\geq 0 \\ x_3 - x_1 &\geq 0 \end{aligned}$$

$$\begin{aligned}
 x_4 - x_1 &\geq 0 \\
 x_3 + x_2 - x_1 - x_4 &\geq 0 \\
 x_4 + x_2 - x_1 - x_3 &\geq 0 \\
 x_4 + x_3 - x_1 - x_2 &\geq 0
 \end{aligned}$$

最小传动角的要求, 应满足下列关系式:

$$2x_2x_3 \cos[\gamma] - x_2^2 - x_3^2 + (x_4 - x_1)^2 \geq 0$$

式中, $[\gamma]$ 为最小传动角。

飞剪机结构布置因素, 各变量应在各变量的上、下限内取值, 即

$$x_{i\min} \leq x_i \leq x_{i\max} \quad (i=1, 2, 3, 4, 5)$$

式中, $x_{i\min}$ 和 $x_{i\max}$ 分别为 x_i 取值的上下限。

综上所述, 飞剪机剪切机构的优化设计数学模型的标准形式为

$$\min f(X) = \min [W_1 f_1(X) + W_2 f_2(X)] \quad (X \in R^5) \quad (8.1.41)$$

满足于

$$\begin{aligned}
 g_1(X) &= x_1 - x_2 \leq 0 \\
 g_2(X) &= x_1 - x_3 \leq 0 \\
 g_3(X) &= x_1 - x_4 \leq 0 \\
 g_4(X) &= x_1 + x_4 - x_2 - x_3 \leq 0 \\
 g_5(X) &= x_1 + x_3 - x_2 - x_4 \leq 0 \\
 g_6(X) &= x_1 + x_2 - x_3 - x_4 \leq 0 \\
 g_7(X) &= x_2^2 + x_3^2 - 2x_2x_3 \cos[\gamma] - (x_4 - x_1)^2 \leq 0 \\
 g_{7+2i-1}(X) &= x_{i\min} - x_i \leq 0 \quad (i=1, 2, \dots, 5) \\
 g_{7+2i}(X) &= x_i - x_{i\max} \leq 0 \quad (i=1, 2, \dots, 5)
 \end{aligned}$$

8.1.4 50t 飞剪机剪切机构的优化设计

对 60 mm × 60 mm 钢坯 50t 飞剪机的剪切机构进行优化设计。根据该机构的具体情况, 取 $200 \leq x_1 \leq 500$, $200 \leq x_2 \leq 1080$, $200 \leq x_3 \leq 550$, $200 \leq x_4 \leq 1050$, $-30^\circ \leq x_5 \leq 30^\circ$,

$V_{x1}=2000\text{mm/s}$, $l_c=1285\text{mm}$, $\xi=30\text{mm}$, $[\gamma]=30^\circ$, $k=6$, $W_1=\frac{1}{200}$, $W_2=10$ 。

用复合形法进行求解, 其优化结果与文献[32]的优化结果和原设计方案的有关数据的比较分别列于表 8.1.1 和表 8.1.2 中。从表 8.1.1 和表 8.1.2 的数据比较可知, 作者在文献[30]中, 对飞剪机剪切机构的优化设计的优化结果, 无论是剪切刀刃水平上的速度均匀程度, 还是连杆 MA 与 x 轴的位置角, 都优于文献[32]中的优化结果和原设计方案。这主要是对剪切机构附加的当连杆 MA 与 y 轴重合时, 曲柄与连杆在同一条铅垂线上和两个剪切刀刃的运动轨迹曲线的重叠度精确地为某一特定值的条件所致。同时, 能使两个剪切刀刃的运动轨迹曲线的重叠度 ξ 精确地为预先给定的某一特定值。

表 8.1.1 优化的飞剪机构尺寸参数和有关数据的比较

| | | 优化的结果 ^[30] | 优化的结果 ^[32] | 原设计方案 ^[32] |
|--------|------------|-----------------------|-----------------------|-----------------------|
| 构件尺寸参数 | l_1 | 370.455 | 330 | 320 |
| | L_2 | 1025.684 | 1020.08 | 1015 |
| | L_3 | 467.122 | 490.40 | 470 |
| | L_4 | 1034.079 | 973.91 | 1050 |
| | L_5 | 207.045 | 327.49 | 360 |
| | α | $-18^\circ 58'55''$ | $-10^\circ 53'9''$ | -18° |
| | θ | $103^\circ 32'14''$ | $91^\circ 39'23''$ | 90° |
| 有关数据比较 | ξ | 30 | 29.98 | 74.61 |
| | $f_1(X^*)$ | 1278.346 | 1626.702 | 2963.415 |
| | $f_2(X^*)$ | 0.325 600 | 1.590 897 | 8.665 169 |
| | $f(X^*)$ | 9.647 725 | 24.042 48 | 101.468 8 |
| | ω | 6.274 073 | 7.139 445 | 8.816 907 |
| | γ_1 | $30^\circ 11''$ | $30^\circ 8''$ | $41^\circ 10'6''$ |

注： γ_1 为剪切机构的最小传动角。

表 8.1.2 剪切机构在剪切区间内有关数据的比较

| | ϕ_i | x_M | y_M | V_x | $\bar{\delta}$ |
|------------------|---------------------|--------|--------|----------|--------------------|
| 文献 [30] 优化 | $80^\circ 26'50''$ | 176.41 | 612.5 | -2000 | $90^\circ 7'00''$ |
| | $86^\circ 9'15''$ | 143.71 | 628.48 | -2114.83 | $90^\circ 6''$ |
| | $91^\circ 51'40''$ | 109.37 | 641.08 | -2207.63 | $89^\circ 56'12''$ |
| | $97^\circ 34'5''$ | 73.74 | 650.17 | -2278.09 | $89^\circ 54'54''$ |
| | $103^\circ 16'30''$ | 37.16 | 655.66 | -2326.08 | $89^\circ 56'8''$ |
| | $108^\circ 58'55''$ | 0 | 657.5 | -2351.71 | 90° |
| 文献 [32] 优化 | $70^\circ 37'7''$ | 168.27 | 612.5 | -2000 | $89^\circ 39'42''$ |
| | $76^\circ 40'15''$ | 137.58 | 628.48 | -2143.73 | $89^\circ 36'38''$ |
| | $82^\circ 43'23''$ | 104.96 | 641.04 | -2260.99 | $89^\circ 38'1''$ |
| | $88^\circ 46'32''$ | 70.80 | 650.14 | -2351.75 | $89^\circ 43'30''$ |
| | $94^\circ 49'40''$ | 35.50 | 655.65 | -2416.06 | $89^\circ 52'56''$ |
| | $100^\circ 52'48''$ | -0.56 | 657.49 | -2454.17 | $90^\circ 6'14''$ |
| 原设计方案 | $70^\circ 6'16''$ | 200.53 | 612.5 | -2000 | $89^\circ 22'3''$ |
| | $77^\circ 41'11''$ | 168.48 | 636.18 | -2262.82 | $88^\circ 53'31''$ |
| | $85^\circ 16'6''$ | 132.85 | 655.03 | -2476.99 | $88^\circ 32'12''$ |
| | $92^\circ 51'1''$ | 94.37 | 668.72 | -2641.90 | $88^\circ 17'25''$ |
| | $100^\circ 25'56''$ | 53.79 | 677.02 | -2757.45 | $88^\circ 8'46''$ |
| | $108^\circ 50''$ | 11.84 | 679.80 | -2824.26 | $88^\circ 6'8''$ |

8.2 蟹爪式装载机扒取机构的优化设计^[33]

8.2.1 扒取机构

蟹爪式装载机是矿山采掘、土方工程、仓库和码头等用于散颗粒物料连续装载的一种高效率设备。在蟹爪式装载机中，它的工作机构如图 8.2.1 所示^[32]。工作机构由两个对称布置的扒取四连杆机构组成。连杆的伸出端呈仿“蟹爪”形状，当曲柄连续转动时，“蟹爪”的端点 M 产生一定形状的运动曲线，模仿蟹爪动作。两个扒取机构的曲柄位置错开 180° ，因而两个扒爪就能交替地进行扒取物料，实现连续装载。

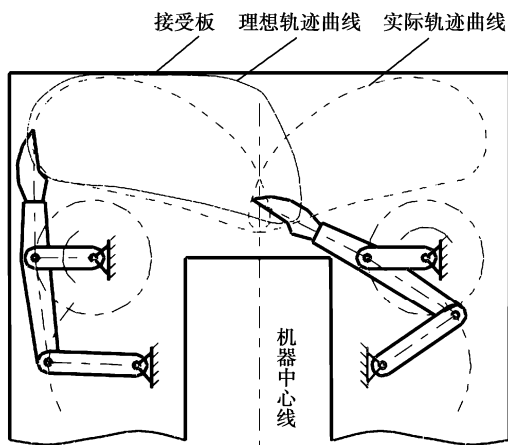


图 8.2.1 蟹爪式装载机扒取机构

从运动学上来说，蟹爪式装载机扒取机构有两种基本结构形式。第一种结构形式是曲柄弧形导杆机构，如图 8.2.2 (a) 所示；第二种结构形式是曲柄直线导杆机构，如图 8.2.2 (b) 所示。不论采用哪种形式的机构，从实现扒取物料的高效率和能量的低消耗来看，对动作都有一定的基本要求。

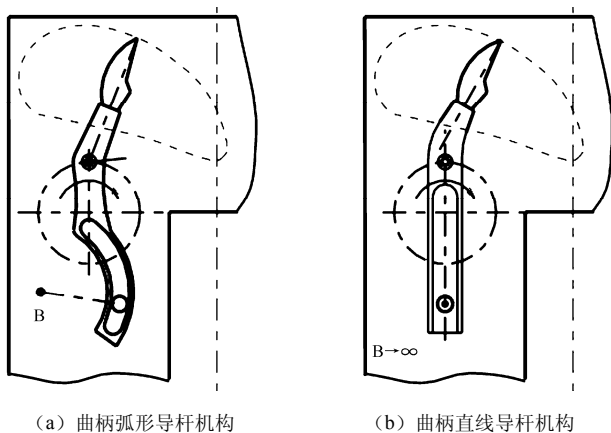


图 8.2.2 蟹爪式装载机扒取机构的形式

扒取机构的一个工作循环分为扒爪插入料堆、扒集物料、扒爪将物料沿接受板送到运输槽和扒爪空回运动 4 个阶段, 对这 4 个阶段的基本要求如下^[32]:

(1) 当扒爪插入料堆时, 扒爪的位置角 (即扒爪的几何中心线与料堆工作面之间的夹角) 最好接近 90° , 这样可以减少作用在扒爪上的阻力, 使扒爪顺利插入料堆。

(2) 当扒爪扒取物料时, 希望扒爪不要过深地插入料堆, 扒爪端点的运动轨迹适当地伸出接受板的前端边缘, 并且最好与边缘平行。同时, 为了有效地扒集物料, 扒爪的位置角最好在 $48^\circ \sim 85^\circ$ 之间变化。

(3) 当扒爪沿着接受板移动物料时, 要求轨迹曲线有一定的长度, 以便能有效地把物料推送到运输槽内。随后, 要求从料堆中扒取出来的物料全部送走, 避免出现物料在左右扒爪之间抛来抛去的现象, 因此, 要求扒爪在这一阶段的位置角不宜过大, 最好在 $23^\circ \sim 60^\circ$ 范围内。

(4) 扒爪返回时的运动轨迹与扒集阶段的运动轨迹之间的距离不能过小, 防止扒爪在空回程时带走大块的物料, 这就要求曲柄半径与最大块度之间有如下关系:

$$l_1 = (0.35 \sim 0.4) D_{\max} \quad (8.2.1)$$

式中, D_{\max} 为物料的最大块度, 以径向的最大尺寸计算; l_1 为曲柄的长度。

8.2.2 扒取机构的运动学分析

扒爪机构简图如图 8.2.3 所示。

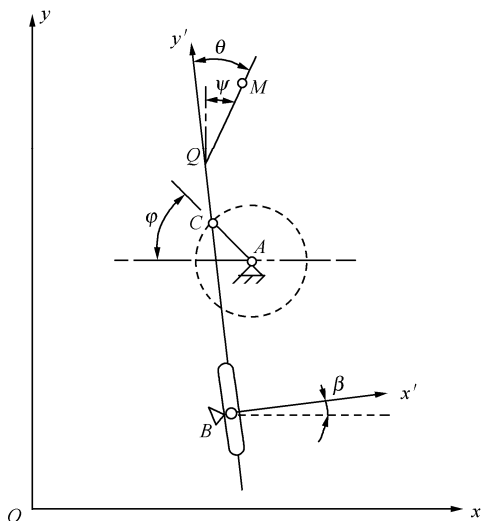


图 8.2.3 扒取机构简图

图 8.2.3 中符号的意义分别是:

A——曲柄回转中心, 坐标为 (x_A, y_A) ;

B——导杆摆动支点, 坐标为 (x_B, y_B) ;

C——曲柄铰链点, 坐标为 (x_C, y_C) ;

M ——导杆扒爪端点, 在动坐标 $x'B'y'$ 中, 坐标为 (x'_M, y'_M) , 在固定坐标 xOy 中, 坐标为 (x_M, y_M) ;

φ ——曲柄转角, 由负 x 轴起计, 顺时针为正;

β ——导杆对 y 轴的偏角, 即动坐标相对于固定坐标的旋转角, 逆时针为正;

θ ——导杆中心线与扒爪中心线的夹角, 由 y' 轴起计, 顺时针为正;

l_1 ——曲柄长度, 即 AC 的长度;

l_2 —— Q 点到 C 点的距离;

l_3 —— M 点到 Q 点的距离;

l ——导杆的变长度, 即 CB 的长度。

由图 8.2.3 的几何关系, 可得

$$x_C = x_A - l_1 \cos \varphi \quad (8.2.2)$$

$$y_C = y_A + l_1 \sin \varphi \quad (8.2.3)$$

$$l = [(x_C - x_B)^2 + (y_C - y_B)^2]^{\frac{1}{2}} \quad (8.2.4)$$

$$\beta = \arctan \frac{l_1 \cos \varphi - x_A + x_B}{y_A - y_B + l_1 \sin \varphi} \quad (8.2.5)$$

$$x'_M = l_3 \sin \theta \quad (8.2.6)$$

$$y'_M = l_3 \cos \theta + l + l_2 \quad (8.2.7)$$

根据坐标系平移和旋转的关系, 扒爪端点 M 的运动轨迹曲线方程为

$$x_M = x'_M \cos \beta - y'_M \sin \beta + x_B \quad (8.2.8)$$

$$y_M = x'_M \sin \beta + y'_M \cos \beta + y_B \quad (8.2.9)$$

$$\psi = \theta - \beta \quad (8.2.10)$$

式中, ψ 为扒爪位置角, 由 y 轴起计, 顺时针为正。

8.2.3 扒取机构优化设计的数学模型的建立

1. 设计变量

扒取机构优化设计变量为曲柄回转中心 A 坐标 (x_A, y_A) 、导杆摆动支点 B 坐标 (x_B, y_B) 、扒取四连杆机构的尺寸参数 $l_i (i=1,2,3)$ 和导杆中心线与扒爪中心线的夹角 θ , 即

$$X = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)^T = (l_1, l_2, l_3, x_A, y_A, x_B, y_B, \theta)^T \quad (8.2.11)$$

2. 目标函数

在文献[32]中, 对蟹爪式装载机进行了优化设计, 其中目标函数是这样选取的, 用作图法确定曲柄 16 个转角 $\varphi_j (j=1,2,\dots,16)$ 对应于理想轨迹曲线上的 16 个点的位置, 再选取这个曲柄 16 个转角 φ_j 对应的扒爪端点的理想位置与实际位置之差的平方之和作为目标函数。由于完全满足理想轨迹曲线的机构不存在, 即使存在, 事先也无法知道, 因此, 文献[32]中确定理想轨迹曲线上的 16 个点的位置存在误差。从而导致目标函数不能真实地反映理想轨迹曲线与实际轨迹曲线之间的误差, 最终导致优化结果不理想, 其优化结果见 8.2.4 节中的优化实例。

下面, 讨论如何合理地建立蟹爪式装载机扒取机构优化设计的目标函数。

从对蟹爪式装载机扒取机构的 4 条要求中, 我们可将它们归纳为两点: 第一点是尽量使导杆扒爪端点 M 的实际运动轨迹曲线与理想轨迹曲线重合; 第二点是尽量使扒爪的实际位置角与理想位置角重合。

为了衡量导杆扒爪端点 M 的实际运动轨迹曲线与理想轨迹曲线的误差, 作者将导杆扒爪端点 M 的实际运动轨迹曲线与理想轨迹曲线分为外曲线和内曲线。外曲线由导杆扒爪端点 M 的实际运动轨迹曲线和理想轨迹曲线外面的曲线部分组成, 内曲线由导杆扒爪端点 M 的实际运动轨迹曲线和理想轨迹曲线里面的曲线部分组成, 如图 8.2.4 所示。

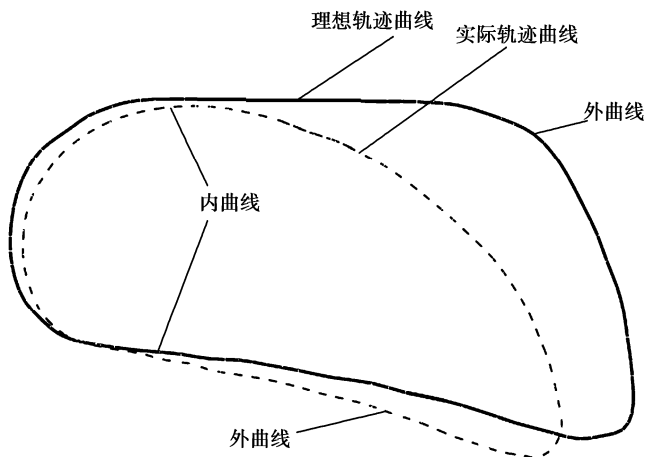


图 8.2.4 轨迹曲线与内、外曲线

导杆扒爪端点 M 的运动轨迹曲线与理想轨迹曲线之间的误差可用如下函数来衡量:

$$f_1(X) = S_{\text{外}} - S_{\text{内}} \quad (8.2.12)$$

式中, $S_{\text{外}}$ 和 $S_{\text{内}}$ 分别为外曲线所包围的面积和内曲线所包围的面积。

下面, 我们来讨论 $S_{\text{外}}$ 和 $S_{\text{内}}$ 的计算问题, 计算时用折线代替曲线。具体计算步骤如下:

第一步, 将理想轨迹曲线用折线代替, 用 k 个点 (如 16 个点) 的折线代替后的理想轨迹曲线如图 8.2.5 所示。

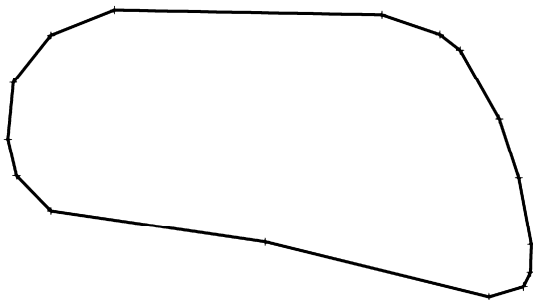


图 8.2.5 用折线近似代替理想轨迹曲线

第二步,按曲柄等分角度计算导杆扒爪端点 M 的位置,将各点连接成封闭曲线来近似代替实际轨迹曲线。

第三步,在理想轨迹曲线内取一点,如图 8.2.6 中的 O 点。为了计算方便,尽量使 O 点在内曲线的几何中心位置。

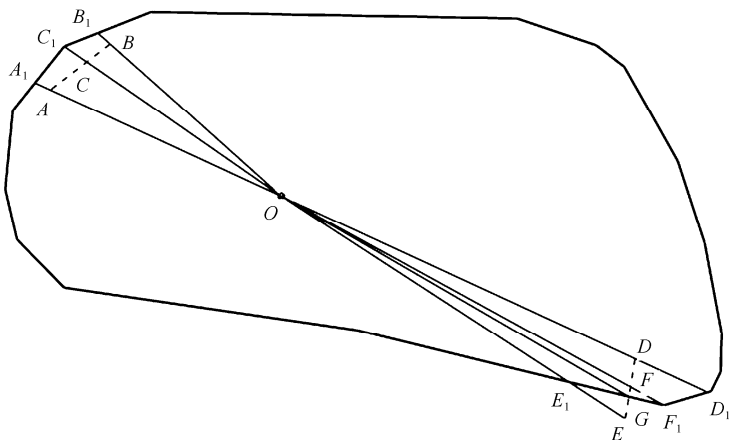


图 8.2.6 内、外曲线面积计算原理图

第四步,按近似实际轨迹曲线各直线段,计算理想轨迹曲线和实际轨迹曲线各三角形的面积。计算时,有下述两种情况发生。

第一种情况是实际轨迹与理想轨迹不相交的情况。假设近似的实际轨迹曲线第 i 段为 AB , AB 段与理想轨迹曲线不相交(见图 8.2.6), OA 和 OB 的延长分别交于理想轨迹曲线于 A_1 点和 B_1 点,由于 A_1 点和 B_1 点分别交于近似理想轨迹的不同直线段上,设 C_1 点为近似理想轨迹的这两条不同直线段的交点。则

$$S_{\text{外}i} = S_{\triangle OC_1A_1} + S_{\triangle OB_1C_1} \quad (8.2.13)$$

$$S_{\text{内}i} = S_{\triangle OBA} \quad (8.2.14)$$

式中, $S_{\text{内}i}$ 为近似的实际轨迹曲线第 i 段为 AB 和内点 O 组成的三角形的面积, $S_{\text{外}i}$ 为近似的实际轨迹曲线第 i 段为 AB 对应的理想轨迹曲线段 $A_1C_1B_1$ 和内点 O 组成的四边形的面积, $S_{\triangle OBA}$ 表示 $\triangle OBA$ 的面积,其余类推。

以 $\triangle OBA$ 为例来说明三角形面积的计算。设 $\triangle OBA$ 的三个顶点的坐标值分别为 O 点的坐标为 (x_o, y_o) , B 点的坐标为 (x_B, y_B) , A 点的坐标为 (x_A, y_A) 。则

$$S_{\triangle OBA} = \frac{1}{2} \begin{vmatrix} 1 & x_o & y_o \\ 1 & x_B & y_B \\ 1 & x_A & y_A \end{vmatrix} = \frac{1}{2} [(x_B - x_o)(y_A - y_o) - (x_A - x_o)(y_B - y_o)] \quad (8.2.15)$$

第二种情况是实际轨迹与理想轨迹相交的情况。假设近似的实际轨迹曲线第 j 段为 DE , DE 段与理想轨迹曲线相交于 G 点(见图 8.2.6), OD 和 OE 的延长分别交于理想轨迹曲线于 D_1 点和 E_1 点,由于 D_1 点和 E_1 点分别交于近似理想轨迹的不同直线段上,设 F_1 点为近似理想轨迹的这两条不同直线段的交点。则

$$S_{外j} = S_{\triangle OF_1D_1} + S_{\triangle OGF_1} + S_{\triangle OEG} \quad (8.2.16)$$

$$S_{内j} = S_{\triangle OGD} + S_{\triangle OE_1G} \quad (8.2.17)$$

第五步, 将第四步按近似实际轨迹曲线各直线段计算的外曲线和内曲线的各三角形的面积 $S_{外i}$ 和 $S_{内i}$ 相加, 即

$$S_{外} = \sum_{i=1}^k S_{外i} \quad (8.2.18)$$

$$S_{内} = \sum_{i=1}^k S_{内i} \quad (8.2.19)$$

用 $f_2(X)$ 来衡量扒爪实际位置角与理想位置角之间的误差函数。为了使 $f_2(X)$ 能真正反映扒爪实际位置角与理想位置角之间的误差函数, 首先, 将实际轨迹曲线的 x 取值区间进行放大或缩小、平移使之与理论轨迹曲线的 x 取值区间相同; 其次, 将轨迹曲线在 x 最大和最小处分为上曲线和下曲线; 最后, 将有位置角要求的上、下曲线的 x 区间段内取 N 个点, 设这些点的 x 坐标为 x_j ($j=1, 2, \dots, N$); 取 $f_2(X)$ 为这 N 个点处实际位置角与理想位置角之差的绝对值之和, 即

$$f_2(X) = \sum_{j=1}^N |\psi'(x_j) - \psi(x_j)| \quad (8.2.20)$$

式中, $\psi(x_j)$ 为 x_j 处的理想位置角, $\psi'(x_j)$ 为与 x_j 处相对应的实际轨迹曲线 x_M 处的实际位置角。因为, 将实际轨迹曲线进行平移和放大, 因此, 作者称 $\psi'(x_j)$ 为相对位置角。根据放大和平移的关系, x_j 与 x_M 有如下关系:

$$x_j = \alpha(x_M - x_{M \min}) + x_{\min} \quad (8.2.21)$$

式中, $\alpha = \frac{x_{\max} - x_{\min}}{x_{M \max} - x_{M \min}}$, x_{\min} 和 x_{\max} 分别为理想轨迹曲线 x 的取值区间的最小值和最大值,

$x_{M \min}$ 和 $x_{M \max}$ 分别为实际轨迹曲线 x 的取值区间的最小值和最大值。

综上所述, 扒取机构的优化是使导杆扒爪端点 M 的运动轨迹曲线与理想轨迹曲线之间的误差、扒爪实际位置角与扒爪理想位置角之间的误差最小。它是一个多目标函数的优化设计问题, 用线性加权法将多目标函数转化为单目标函数, 则扒取机构的优化设计的目标函数为

$$f(X) = W_1 f_1(X) + W_2 f_2(X) \quad (8.2.22)$$

式中, W_1 和 W_2 是加权因子, $f_1(X)$ 为扒爪端点 M 实际运动轨迹曲线与理想轨迹曲线之间的误差函数, $f_2(X)$ 为扒爪实际位置角与理想位置角之间的误差函数。

3. 约束条件

根据扒取机构的运动学特性、工艺条件和机器的总体设计要求, 建立如下约束条件:

$$\begin{aligned} x_1 - 0.35D_{\max} &\geq 0 \\ \left[(x_4 - x_6)^2 + (x_5 - x_7)^2 \right]^{\frac{1}{2}} - x_1 &\geq 0 \\ x_2 &\geq 0 \\ x_3 &\geq 0 \end{aligned}$$

$$x_{A\min} \leq x_4 \leq x_{A\max} \quad (8.2.23)$$

$$y_{A\min} \leq x_5 \leq y_{A\max}$$

$$x_{B\min} \leq x_6 \leq x_{B\max}$$

$$y_{B\min} \leq x_7 \leq y_{B\max}$$

式中, D_{\max} 为物料的最大块度, 以径向最大尺寸计算; $x_{A\min}$ 表示 x_A 的最小取值, 其余类推。

综上所述, 蟹爪式装载机扒取机构优化设计数学模型的标准形式为

$$\min f(X) = \min [W_1 f_1(X) + W_2 f_2(X)] \quad (X \in R^8) \quad (8.2.24)$$

满足于

$$\begin{aligned} g_1(X) &= 0.35D_{\max} - x_1 \leq 0 \\ g_2(X) &= x_1 - [(x_4 - x_6)^2 + (x_5 - x_7)^2]^{\frac{1}{2}} \leq 0 \\ g_3(X) &= -x_2 \leq 0 \\ g_4(X) &= -x_3 \leq 0 \\ g_5(X) &= x_{A\min} - x_4 \leq 0 \\ g_6(X) &= x_4 - x_{A\max} \leq 0 \\ g_7(X) &= y_{A\min} - x_5 \leq 0 \\ g_8(X) &= x_5 - y_{A\max} \leq 0 \\ g_9(X) &= x_{B\min} - x_6 \leq 0 \\ g_{10}(X) &= x_6 - x_{B\max} \leq 0 \\ g_{11}(X) &= y_{B\min} - x_7 \leq 0 \\ g_{12}(X) &= x_7 - y_{B\max} \leq 0 \end{aligned}$$

8.2.4 矿用蟹爪式装载机的扒取机构优化设计实例

以 2100mm 机宽的矿用蟹爪式装载机的扒取机构为例进行优化设计。已知机头接受板宽为 2100mm, 运输槽宽为 700mm, 运输槽至接受板的前端距离为 900mm, 两曲柄回转中心距离为 1460mm, 根据传动装置的尺寸, 曲柄回转中心 A 点距接受板前端距离为 780mm, 装载最大块度 $D_{\max} = 600\text{mm}$ 。

部分变量的上、下限为: $280 \leq x_A \leq 400$, $380 \leq y_A \leq 470$, $280 \leq x_B \leq 370$, $0 \leq y_B \leq 60$ 。理想轨迹曲线部分点的坐标及理想位置角如表 8.2.1 所示。作者在优化设计计算中, 为了减少 $f_1(X)$ 近似计算带来的误差, 在理想轨迹曲线上附加了 8 个点, 这些附加点是根据文献[32]的理想轨迹曲线图得到的。取加权因子 $W_1 = 1/1000$, $W_2 = 1/10$, 用复合形法求得蟹爪式装载机扒取机构最优解为 (括号中的数据为文献[32]的最优解):

$$\begin{aligned} x_1^* &= l_1 = 230.070 & (224.930 \text{ mm}) \\ x_2^* &= l_2 = 341.782 & (110.055 \text{ mm}) \\ x_3^* &= l_3 = 362.162 & (539.556 \text{ mm}) \\ x_4^* &= x_A = 395.776 & (319.761 \text{ mm}) \\ x_5^* &= y_A = 407.002 & (419.968 \text{ mm}) \\ x_6^* &= x_B = 369.957 & (314.776 \text{ mm}) \end{aligned}$$

$$x_7^* = y_B = 2.744 \quad (20.330^\circ)$$

$$x_8^* = \theta = 31.194^\circ \quad (35.083^\circ)$$

$$f(X^*) = 65.91428$$

作者优化后的实际轨迹曲线与理想轨迹曲线如图 8.2.7 所示, 理想轨迹曲线与位置角和优化后相对位置角如表 8.2.1 所示, 曲柄各转角与实际轨迹曲线点及位置角如表 8.2.2 所示(表中括号内的数据为文献[32]的数据)。

对图 8.2.7、表 8.2.1 和表 8.2.2 进行比较发现, 文献[33]优化后的位置角与文献[32]优化后的位置角基本相同, 并且与理论位置角相差不大。但文献[33]优化后的轨迹曲线更接近于理想轨迹曲线, 这主要是 $f_1(X)$ 更能反映实际轨迹与理想轨迹的误差, 因此, 作者所用的目标函数为诸如此类优化问题较为理想的目标函数。

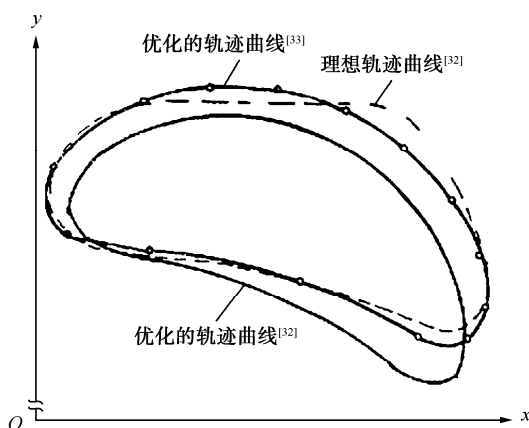


图 8.2.7 理想轨迹曲线与优化轨迹曲线

表 8.2.1 理想轨迹曲线、位置角和优化机构的相对位置角

| 点 序 号 | 理想轨迹曲线位置 | | 理想位置角 | 优化相对位置角 |
|-------------|------------|------------|--------------------------|---------------------------|
| | x_j (mm) | y_j (mm) | $\psi(x_j)$ ($^\circ$) | $\psi'(x_j)$ ($^\circ$) |
| 1 | 45 | 1090 | 5 | 4.570 (6.060) |
| 2 | 150 | 1200 | 10 | 11.491 (12.853) |
| 3 | 300 | 1240 | 20 | 19.219 (20.400) |
| 4 | 570 | 1240 | 30 | 31.486 (32.374) |
| 5 | 830 | 1240 | 40 | 43.410 (44.023) |
| 6 | 930 | 1220 | 50 | 48.514 (49.015) |
| 7 | 1000 | 1150 | 60 | 52.486 (52.910) |
| 8 | 1050 | 1070 | 65 | 55.676 (55.047) |
| 9 | 1160 | 820 | 70 | 67.394 (67.699) |

续表

| 点 序 号 | 理想轨迹曲线位置 | | 理想位置角 | 优化相对位置角 |
|-------------|------------|------------|-----------------|------------------|
| | x_j (mm) | y_j (mm) | $\psi(x_j)$ (°) | $\psi'(x_j)$ (°) |
| 10 | 1120 | 700 | | |
| 11 | 1000 | 680 | | |
| 12 | 900 | 720 | | |
| 13 | 480 | 830 | | |
| 14 | 280 | 850 | | |
| 15 | 110 | 880 | | |
| 16 | 20 | 980 | | |

表 8.2.2 优化机构的曲柄转角与轨迹曲线

| 曲柄转角 | x_M (mm) | y_M (mm) | 位置角 (°) |
|-------|---------------------|---------------------|-----------------|
| 0 | 39.291 (100.571) | 1073.158 (1052.729) | 4.389 (6.256) |
| 22.5 | 127.514 (185.584) | 1170.799 (1174.533) | 10.421 (12.419) |
| 45 | 262.526 (313.042) | 1247.092 (1193.196) | 17.622 (19.666) |
| 67.5 | 428.962 (468.092) | 1286.678 (1215.554) | 25.433 (27.479) |
| 90 | 609.698 (634.276) | 1280.486 (1193.988) | 33.525 (35.540) |
| 112.5 | 786.563 (794.367) | 1226.264 (1127.287) | 41.653 (43.809) |
| 135 | 942.036 (931.762) | 1128.805 (1020.894) | 49.586 (51.446) |
| 157.5 | 1061.124 (1032.968) | 999.749 (886.823) | 57.031 (58.742) |
| 180 | 1132.833 (1088.562) | 857.234 (747.425) | 63.527 (63.795) |
| 202.5 | 1150.346 (1093.915) | 726.348 (616.153) | 68.205 (69.245) |
| 225 | 1107.186 (1045.767) | 642.620 (541.951) | - |
| 247.5 | 977.839 (922.537) | 658.869 (577.456) | - |
| 270 | 676.862 (645.490) | 793.969 (737.550) | - |
| 292.5 | 284.977 (304.582) | 872.108 (840.964) | - |
| 315 | 75.105 (124.421) | 903.700 (892.663) | - |
| 337.5 | 13.408 (72.884) | 975.395 (965.463) | - |

8.3 热锯机送进机构的优化设计^[34]

8.3.1 热锯机四连杆送进机构

热锯机是广泛应用于冶金厂型钢、尤其是异型断面型钢轧制生产线上的一种切断设备，它的主要作用是将轧机轧制出来的轧件切头、切尾和切定尺。由于热锯机的进锯速度远小于锯齿线速度，所以每个锯齿只能刮下极薄的一层金属，这是其他切断方法所无法比拟的。另外，与其他切断设备比较，热锯机具有设备简单、重量轻、生产效率高的优点。热锯机一般装设在轧机后面的生产线上，在很多情况下，整个轧钢车间的生产量常因热锯机生产能力的限制而受到影响。

四连杆机构作为热锯机的送进机构，由于机构简单、运转灵活和维修方便，所以应用广泛。图 8.3.1 为 $\phi 1800$ 四连杆式热锯机的结构图^[31]。

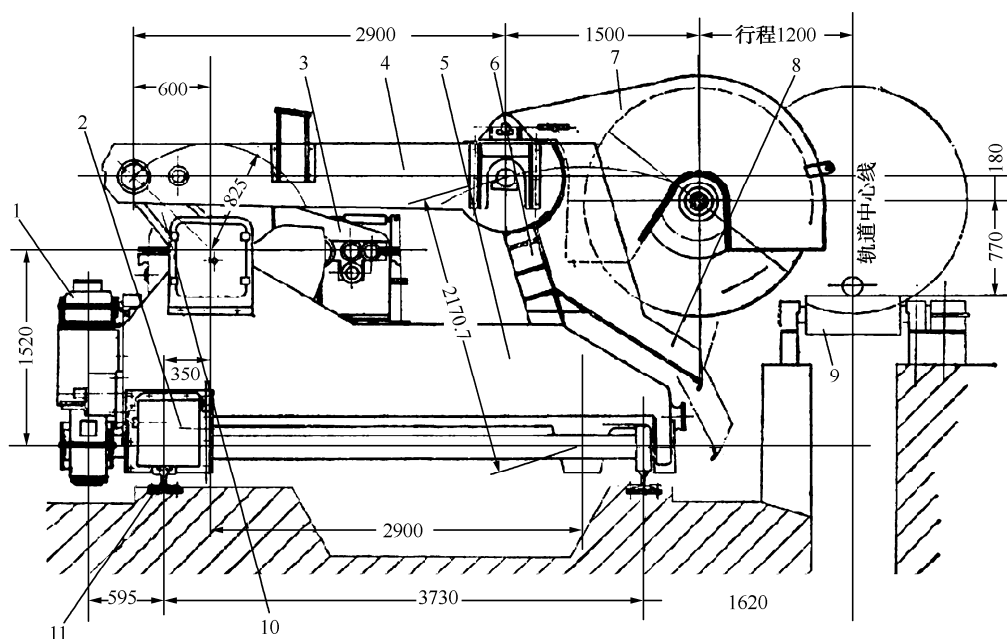


图 8.3.1 $\phi 1800$ 四连杆式热锯机

- 1—横移减速机 2—夹轨器 3—送进减速机 4—锯架 5—锯座
6—摇杆 7—锯片罩 8—防护罩 9—辊道 10—曲柄 11—轨道

8.3.2 送进机构运动学分析

热锯机四连杆送进机构如图 8.3.2 所示, 图中符号的意义分别是:

O_1 ——曲柄回转中心, 坐标为 $(0, y_{o1})$;

O_2 ——摆杆回转中心, 坐标为 $(x_{o2}, 0)$;

A ——曲柄铰链点;

B ——摆杆铰链点;

D ——锯轴中心点;

φ_1 ——曲柄 O_1A 与负 x 轴的夹角, 由负 x 轴起计, 顺时针为正;

φ_2 ——连杆 AB 与负 x 轴的夹角, 由负 x 轴起计, 顺时针为正;

φ_3 ——摆杆 O_2B 与负 x 轴的夹角, 由负 x 轴起计, 顺时针为正;

l_1 ——曲柄长度, 即 O_1A 杆的长度;

l_2 ——连杆长度, 即 AB 杆的长度;

l_3 ——摆杆长度, 即 O_2B 杆的长度;

l_4 —— B 点到 C 点的距离, 即 BC 杆的长度;

l_5 —— C 点到 D 点的距离, 即 CD 杆的长度。

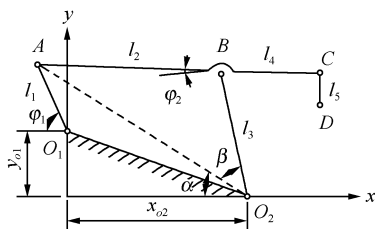


图 8.3.2 热锯机送进机构简图

由图 8.3.2 中的几何关系可得

$$\varphi_3 = \alpha + \beta \quad (8.3.1)$$

$$\varphi_2 = \arctan \frac{y_{o1} + l_1 \sin \varphi_1 - l_3 \sin \varphi_3}{x_{o2} + l_1 \cos \varphi_1 - l_3 \cos \varphi_3} \quad (8.3.2)$$

$$\alpha = \arctan \frac{y_{o1} + l_1 \sin \varphi_1}{x_{o2} + l_1 \cos \varphi_1} \quad (8.3.3)$$

$$\beta = \arccos \frac{C^2 + l_3^2 - l_2^2}{2Cl_3} \quad (8.3.4)$$

式中

$$C = \sqrt{(y_{o1} + l_1 \sin \varphi_1)^2 + (x_{o2} + l_1 \cos \varphi_1)^2}$$

由于 $\angle BCD=90^\circ$ ，由图 8.3.2 中的几何关系，可得锯轴中心 D 的坐标为

$$x_D=(l_4+l_2)\cos\varphi_2-l_1\cos\varphi_1-l_5\sin\varphi_2 \quad (8.3.5)$$

$$y_D=l_3\sin\varphi_3-l_4\sin\varphi_2-l_5\cos\varphi_2 \quad (8.3.6)$$

由上面两式，可得锯轴中心点 D 的速度为

$$V=\sqrt{V_{Dx}^2+V_{Dy}^2} \quad (8.3.7)$$

式中

$$V_{Dx}=-(l_2+l_4)\omega_2\sin\varphi_2+l_1\omega_1\sin\varphi_1-l_5\omega_2\cos\varphi_2$$

$$V_{Dy}=l_3\omega_3\cos\varphi_3-l_4\omega_2\cos\varphi_2+l_5\omega_2\sin\varphi_2$$

而

$$\omega_3=\frac{l_1\omega_1[\sin\varphi_1(x_{o2}-l_3\cos\varphi_3+l_1\cos\varphi_1)+\cos\varphi_1(l_3\sin\varphi_3-y_{o1}-l_1\sin\varphi_1)]}{l_3[\sin\varphi_3(x_{o2}-l_3\cos\varphi_3+l_1\cos\varphi_1)+\cos\varphi_3(l_3\sin\varphi_3-y_{o1}-l_1\sin\varphi_1)]}$$

$$\omega_2=\frac{\cos^2\varphi_2[l_1\omega_1(\cos\varphi_1+\tan\varphi_2\sin\varphi_1)-l_3\omega_3(\cos\varphi_3+\tan\varphi_2\sin\varphi_3)]}{x_{o2}+l_1\cos\varphi_1-l_3\cos\varphi_3}$$

式中， ω_1 为曲柄的角速度。

8.3.3 送进机构优化设计数学模型的建立

1. 设计变量

送进机构的设计变量为各杆的长度、曲柄回转中心 O_1 点和摆杆回转中心 O_2 点的位置，因此，热锯机送进机构优化设计变量为

$$X=(x_1, x_2, \dots, x_7)^T=(l_1, l_2, l_3, l_4, l_5, x_{o2}, y_{o1})^T \quad (8.3.8)$$

2. 目标函数

根据送进机构和热锯机的工作特点， φ_1 在区间 $[60^\circ, 120^\circ]$ 作为热锯机的工作行程阶段是比较理想的。热锯机送进机构优化设计的目标函数可根据对热锯机的要求来确定。对热锯机的要求是：在工作行程阶段，锯轴中心运动速度尽量均匀、锯轴中心运动轨迹在垂直方向波动量尽量小，同时，使热锯机的工作行程为预先给定的值和位置。分别用函数 $f_1(X)$ 来表示锯轴中心运动速度的均匀程度、用函数 $f_2(X)$ 来表示锯轴中心运动轨迹在垂直方向波动量程度、用 $f_3(X)$ 来表示热锯机的实际工作行程值和位置与预先确定工作行程值和位置之差。热锯机送进机构优化设计是一个多目标函数的优化设计问题，用线性加权法将多目标函数转化为单目标函数，则热锯机送进机构优化设计的目标函数为

$$f(X) = W_1 f_1(X) + W_2 f_2(X) + W_3 f_3(X) \quad (8.3.9)$$

式中

$$f_1(X) = \left\{ \frac{1}{m} \sum_{i=1}^m [\bar{V}(\varphi_i) - \bar{V}_m]^2 \right\}^{\frac{1}{2}}$$

$$f_2(X) = \left\{ \frac{1}{m} \sum_{i=1}^m [y_D(\varphi_i) - y_m]^2 \right\}^{\frac{1}{2}}$$

$$f_3(X) = |x_D(60^\circ) - x_{D1}| + |x_D(120^\circ) - x_{D2}|$$

这里

$$\bar{V}_m = \frac{1}{m} \sum_{i=1}^m \bar{V}(\varphi_i)$$

而

$$\bar{V}(\varphi_i) = \frac{V(\varphi_i)}{\omega_1}$$

式中, m 为热锯机工作行程阶段 $\varphi_i = 60^\circ \sim 120^\circ$ 的等分点数; φ_i 为第 i 个等分点曲柄 l_1 所在的位置角; y_m 为热锯机工作行程阶段要求锯轴中心 y_D 的值; x_{D1} 和 x_{D2} 由热锯机的工作行程和位置决定; W_1 、 W_2 和 W_3 为加权系数。

3. 约束条件

根据四杆机构为曲柄摆杆机构的条件、传动角的要求以及热锯机总体的布局等, 可建立如下约束条件:

$$\begin{aligned} x_2 - x_1 &\geq 0 \\ x_3 - x_1 &\geq 0 \\ A - x_1 &\geq 0 \\ x_2 + x_3 - x_1 - A &\geq 0 \\ A + x_3 - x_1 - x_2 &\geq 0 \\ A + x_2 - x_1 - x_3 &\geq 0 \\ \arccos \frac{x_2^2 + x_3^2 - (A - x_1)^2}{2x_2x_3} &\geq \gamma_{\min} \\ \arccos \frac{x_2^2 + x_3^2 - (A + x_1)^2}{2x_2x_3} &\leq \gamma_{\max} \\ x_{i\min} &\leq x_i \leq x_{i\max} \quad (i = 1, 2, \dots, 7) \end{aligned} \quad (8.3.10)$$

式中, $A=\sqrt{x_6^2+x_7^2}$; γ_{\min} 和 γ_{\max} 为 $\angle ABO_2$ 的最小值和最大值(即 $\angle ABO_2$ 的取值范围为 $[\gamma_{\min}, \gamma_{\max}]$); $x_{i\min}$ 和 $x_{i\max}$ 为 x_i 的最大和最小值。

综上所述, 热锯机送进机构的优化设计数学模型的标准形式为

$$\min f(X)=\min[W_1f_1(X)+W_2f_2(X)+W_3f_3(X)] \quad (X \in R^7) \quad (8.3.11)$$

满足于

$$g_1(X)=x_1-x_2 \leq 0$$

$$g_2(X)=x_1-x_3 \leq 0$$

$$g_3(X)=x_1-A \leq 0$$

$$g_4(X)=x_1+A-x_2-x_3 \leq 0$$

$$g_5(X)=x_1+x_2-A-x_3 \leq 0$$

$$g_6(X)=x_1+x_3-A-x_2 \leq 0$$

$$g_7(X)=\gamma_{\min}-\arccos \frac{x_2^2+x_3^2-(A-x_1)^2}{2x_2x_3} \leq 0$$

$$g_8(X)=\arccos \frac{x_2^2+x_3^2-(A+x_1)^2}{2x_2x_3}-\gamma_{\max} \leq 0$$

$$g_{8+2i-1}(X)=x_{i\min}-x_i \leq 0 \quad (i=1,2,\dots,7)$$

$$g_{8+2i}(X)=x_i-x_{i\max} \leq 0 \quad (i=1,2,\dots,7)$$

8.3.4 热锯机送进机构优化设计实例

作者选用文献[35]提供的实例作为优化设计的数例, 取 $W_1=W_2=5$, $W_3=1$, $\gamma_{\max}=140^\circ$, $\gamma_{\min}=40^\circ$, $m=7$, $y_m=2380\text{mm}$, $x_{D1}=4150\text{mm}$, $x_{D2}=5050\text{mm}$, $800 \leq x_1 \leq 1000$, $2800 \leq x_2 \leq 3300$, $2400 \leq x_3 \leq 2800$, $1400 \leq x_4 \leq 1800$, $170 \leq x_5 \leq 210$, $2800 \leq x_6 \leq 3300$, $1600 \leq x_7 \leq 2000$ 。

用复合形法求解, 作者优化的结果为

$$X^*=(900, 3077.527, 2547.594, 1518.521, 190.625, 3075.277, 1600.016)^T$$

文献[35]优化的结果为

$$X^*=(880.434, 2999.954, 2599.96, 1483.92, 181.259, 2999.93, 1800)^T$$

原设计方案为

$$X=(900, 3000, 2610, 1600, 190, 3000, 1800)^T$$

热锯机有关数据的比较如表 8.3.1 所示, 从表中可发现, 作者在文献[34]中优化的机构, 在工作行程阶段, 锯轴中心运动轨迹在垂直方向波动量小, 运动速度均匀程度都优于文献[35]的优化结果和原设计方案; 并且热锯机的工作行程和工作行程的位置精确地为预先给定的值。

表 8.3.1 原机构与优化机构有关数据的比较

| | 文献[34] | | 文献[35] | | 原方案 | |
|-----------------|-----------|-----------|-----------|-----------|-----------|-----------|
| φ_i | y_D | \bar{V} | y_D | \bar{V} | y_D | \bar{V} |
| 60° | 2380.171 | 779.564 | 2381.140 | 744.111 | 2376.374 | 758.822 |
| 70° | 2379.625 | 841.329 | 2378.973 | 810.386 | 2372.959 | 826.905 |
| 80° | 2380.216 | 882.805 | 2378.765 | 857.285 | 2372.080 | 875.544 |
| 90° | 2380.467 | 899.975 | 2378.959 | 880.470 | 2372.085 | 900.051 |
| 100° | 2379.898 | 889.845 | 2378.911 | 876.853 | 2372.281 | 897.149 |
| 110° | 2379.216 | 850.161 | 2379.081 | 844.265 | 2373.136 | 864.596 |
| 120° | 2380.395 | 779.526 | 2381.127 | 781.479 | 2376.380 | 801.217 |
| S | 900 | | 880.433 | | 900.001 | |
| x_{D1} | 4150 | | 4043.654 | | 4149.442 | |
| δ | 0.07876 | | 0.1011 | | 0.1034 | |
| γ_{\min} | 53.28717° | | 55.18957° | | 54.65391° | |
| γ_{\max} | 101.4242° | | 102.6479° | | 103.0473° | |

注：① $\delta = \max \frac{\bar{V}_m - \bar{V}(\varphi_i)}{\bar{V}_m}$ ；

② 工作行程 $S = x_D(120^\circ) - x_D(60^\circ)$ 。

8.4 J88 型冷挤压机拉力肘杆机构优化设计^[36~41]

8.4.1 冷挤压机概述

为了保证顺利地实施冷挤压生产加工，除了具有一副符合技术要求的冷挤压模具以外，还应有适合生产要求的冷挤压设备。

在冷挤压生产中，单位挤压力很大，冷挤压件的精度要求高。因此，对冷挤压机除要求能量大、刚性好、导向精度高以外，还要求冷挤压机能提供适合的挤压速度。为了降低挤压过程中的冲击作用，一般要求冷挤压机具有较高的空程向下和回程速度，而挤压行程时，速度尽量地低和均匀。一般认为较好的挤压速度在 100~400mm/s 范围内^[37]。

J88 型冷挤拉力肘杆机构由曲柄滑块机构加上曲柄摆杆机构组成，如图 8.4.1 所示。由于这两套机构均在接近下死点附近区域内工作，这种压机滑块的工作行程速度缓慢，有利于冷挤压工艺。又由于机构具有急回特性，对提高生产效率有利。但这种压机的缺点是滑块的压力行程较小。有色金属冷挤压时，由于其变形程度较大，不需要很大的工作行程即可挤出较长的零件，因此，采用拉力肘杆式挤压机挤压有色金属是比较理想的。

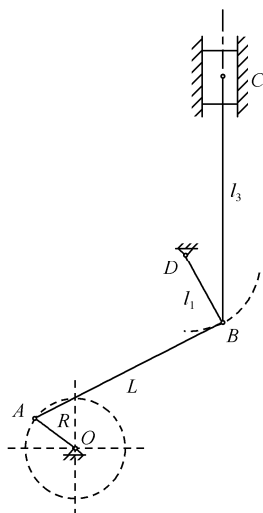


图 8.4.1 拉力肘杆机构

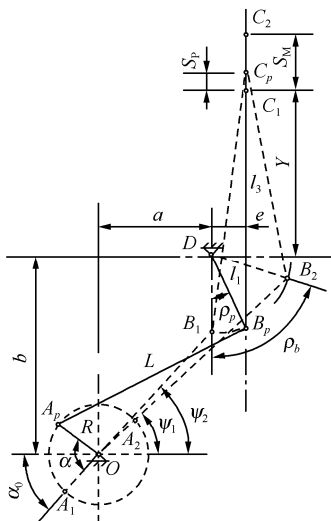


图 8.4.2 拉力肘杆机构参数

8.4.2 J88 型冷挤压机拉力肘杆机构

J88 型冷挤压机拉力肘杆机构由曲柄 OA 、连杆 AB 、摆杆 DB 和拉杆 BC 组成。J88 型冷挤压机拉力肘杆机构如图 8.4.1 和图 8.4.2 所示，图中符号的意义分别是：

- O ——曲柄回转中心；
- D ——摆杆回转中心；
- A ——曲柄铰链点；
- B ——摆杆铰链点；
- C ——拉杆与滑块的铰链点；
- R ——曲柄的长度；
- L ——连杆的长度；
- l_1 ——摆杆的长度；
- l_3 ——拉杆的长度；
- e ——滑块偏置值；
- a ——曲柄回转中心与摆杆回转中心的水平尺寸；
- b ——曲柄回转中心与摆杆回转中心的垂直尺寸。

8.4.3 拉力肘杆机构非独立构件尺寸参数的计算

拉力肘杆机构由 l_1 、 l_3 、 e 、 R 、 L 、 a 和 b 等参数确定。将 l_1 和 a 定为独立参数， l_3 、 e 、 R 、 L 和 b 等为非独立参数。非独立参数由 J88 型冷挤压机的滑块行程 S_M 、滑块的工作行程 S_p 和机构的急回特性系数 K 等技术参数，以及为了防止滑块在一个工作行程中出现两次下死点的

现象而规定压机开始工作时拉杆 l_3 处于铅垂线的位置来确定。

下面, 我们来讨论拉力肘杆机构非独立参数的计算。

由于摆杆、拉杆和滑块组成的机构当摆杆和拉杆共线时出现下死点, 因此, 为了防止滑块在一个工作行程中出现两次下死点的现象, 可取摆杆 l_1 与铅垂线重合时作为滑块的下死点。经过分析可知, 压机开始工作时, 拉杆 l_3 处于铅垂线的位置 (如图 8.4.3 所示的位置) 较为理想^[40]。

由图 8.4.3 中的几何关系, 有

$$l_3 = Y + S_p + \sqrt{l_1^2 - e^2} \quad (8.4.1)$$

$$S_p = l_3 - \sqrt{l_3^2 - e^2} - \sqrt{l_1^2 - e^2} + l_1 \quad (8.4.2)$$

由式 (8.4.1) 和式 (8.4.2) 求解可得

$$e = \sqrt{D_1} \quad (8.4.3)$$

式中

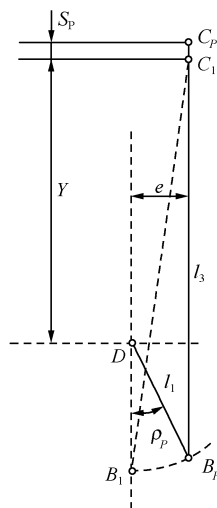


图 8.4.3 机构参数与工作行程关系图

$$D_1 = \frac{-B_1 + \sqrt{B_1^2 - 4C_1}}{2}$$

$$A_1 = Yl_1 - (Y + \frac{S_p}{2})S_p$$

$$B_1 = 2A_1 + (Y + S_p)^2$$

$$C_1 = A_1^2 - (Y + S_p)^2 l_1^2$$

Y 由摆杆回转中心至工作台面的距离、压机的最大封闭高度和拉杆与滑块的铰链点 C 至滑块底面的距离来决定。将偏置 e 代入式 (8.4.1) 可得拉杆的长度 l_3 。

由图 8.4.2 中的几何关系, 可得

$$S_M = \sqrt{l_3^2 - (l_1 \sin \rho_b - e)^2} - l_1 \cos \rho_b - \sqrt{l_3^2 - e^2} + l_1 \quad (8.4.4)$$

解上式得

$$\rho_b = \arcsin \frac{-D_2 + \sqrt{D_2^2 - 4C_2E_2}}{2C_2} \quad (8.4.5)$$

式中

$$A_2 = S_M + \sqrt{l_3^2 - e^2} - l_1$$

$$B_2 = A_2^2 + l_1^2 + e^2 - l_3^2$$

$$C_2 = 4l_1^2(e^2 + A_2^2)$$

$$D_2 = -4B_2l_1e$$

$$E_2 = B_2^2 - 4A_2^2 l_1^2$$

设 ψ 为曲柄和连杆两个极限位置的夹角, 它与机构的急回特性系数 K 的关系为

$$\psi = 180^\circ \cdot \frac{K-1}{K+1} \quad (8.4.6)$$

式中 $K = \frac{V_2}{V_1}$, V_1 、 V_2 分别为滑块回程和滑块向下运行的平均速度。

由图 8.4.2 中的几何关系, 还可得

$$\psi = \psi_2 - \psi_1 = \arctan \frac{b-l_1}{a} - \arctan \frac{b-l_1 \cos \rho_b}{a+l_1 \sin \rho_b} \quad (8.4.7)$$

解上式得

$$b = \frac{-C_3 - \sqrt{C_3^2 - 4B_3 D_3}}{2B_3} \quad (8.4.8)$$

式中

$$A_3 = a + l_1 \sin \rho_b$$

$$B_3 = \tan \psi$$

$$C_3 = -l_1 [B_3 (1 + \cos \rho_b) + \sin \rho_b]$$

$$D_3 = B_3 (l_1^2 \cos \rho_b + a A_3) + l_1 (A_3 - a \cos \rho_b)$$

连杆长度 L 和曲柄半径 R 可分别计算如下:

$$L = \frac{1}{2} \left[\sqrt{a^2 + (b-l_1)^2} + \sqrt{(a+l_1 \sin \rho_b)^2 + (b-l_1 \cos \rho_b)^2} \right] \quad (8.4.9)$$

$$R = L - \sqrt{a^2 + (b-l_1)^2} \quad (8.4.10)$$

8.4.4 拉力肘杆机构的运动分析

取滑块下死点 C_1 作为滑块位移 S 的计算起点, 由图 8.4.4 中的几何关系, 得

$$S = \sqrt{l_3^2 - (l_1 \sin \rho - e)^2} - l_1 \cos \rho - Y \quad (8.4.11)$$

式中

$$\rho = \arccos \frac{C_4^2 + l_1^2 - L^2}{2C_4 l_1} - \arctan \frac{A_4}{B_4}$$

$$A_4 = a + R \cos \delta$$

$$B_4 = b - R \sin \delta$$

$$C_4 = \sqrt{A_4^2 + B_4^2}$$

$$\delta = \alpha - \alpha_0$$

$$\alpha_0 = \arctan \frac{b - l_1}{a}$$

根据拉力肘杆机构的运动特点, 滑块的运动速度为

$$V = \frac{dS}{dt} = \frac{dS}{d\rho} \cdot \frac{d\rho}{d\alpha} \cdot \frac{d\alpha}{dt} \quad (8.4.12)$$

由式 (8.4.11), 可得

$$\frac{dS}{d\rho} = l_1 \sin \rho + \frac{(e - l_1 \sin \rho) \cdot l_1 \cos \rho}{\sqrt{l_3^2 - (e - l_1 \sin \rho)^2}} = l_1 \sin \rho + \frac{l_1 \cos \rho \sin \theta}{\cos \theta} = \frac{l_1 \sin(\rho + \theta)}{\cos \theta} \quad (8.4.13)$$

由图 8.4.4 中的几何关系, 可得

$$L \sin \beta = b - R \sin \delta - l_1 \cos \rho \quad (8.4.14)$$

$$L \cos \beta = a + R \cos \delta + l_1 \sin \rho \quad (8.4.15)$$

分别对式 (8.4.14) 和式 (8.4.15) 两边求导, 整理可得

$$\frac{d\beta}{d\alpha} = \frac{1}{L \cos \beta} \left(-R \cos \delta + l_1 \sin \rho \frac{d\rho}{d\alpha} \right) \quad (8.4.16)$$

$$\frac{d\beta}{d\alpha} = \frac{1}{L \sin \beta} \left(R \sin \delta - l_1 \cos \rho \frac{d\rho}{d\alpha} \right) \quad (8.4.17)$$

由式 (8.4.16) 和式 (8.4.17), 可得

$$\frac{d\rho}{d\alpha} = \frac{R \sin(\delta + \beta)}{l_1 \cos(\rho - \beta)} \quad (8.4.18)$$

将式 (8.4.13) 和式 (8.4.18) 代入式 (8.4.12), 整理可得滑块的速度

$$V = \frac{ds}{dt} = \frac{R\omega \sin(\rho + \theta) \sin(\delta + \beta)}{\cos \theta \cos(\rho - \beta)} \quad (8.4.19)$$

式中

$$\beta = \arctan \frac{b - R \sin \delta - l_1 \cos \rho}{a + R \cos \delta + l_1 \sin \rho}$$

$$\theta = \arcsin \frac{e - l_1 \sin \rho}{l_3}$$

$$\omega = \frac{d\alpha}{dt}$$

ω 为曲柄的角速度。

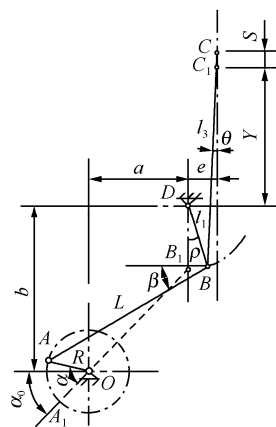


图 8.4.4 拉力肘杆机构运动分析图

8.4.5 拉力肘杆机构优化设计数学模型的建立

1. 设计变量

在 8.4.3 节中, 拉力肘杆机构构件参数 l_1 、 l_3 、 e 、 R 、 L 、 a 和 b 分为独立参数和非独立参数。非独立参数由独立参数和冷挤压机的滑块行程 S_M 、滑块的工作行程 S_P 和机构的急回特性系数 K 等技术参数确定。拉力肘杆机构的优化设计的设计变量由机构的独立参数组成, 因此, 拉力肘杆机构优化设计变量为

$$X=(x_1, x_2)^T=(l_1, a)^T \quad (8.4.20)$$

2. 目标函数

挤压工艺要求滑块的挤压速度尽量均匀, 一般认为较为理想的挤压速度为 $100 \sim 400 \text{ mm/s}$; 挤压速度由滑块 C 来实现, 可用如下函数来衡量滑块挤压速度均匀的程度:

$$f_1(X)=\left\{\sum_{i=1}^M[\bar{V}(X, \delta_i)-\bar{V}_m]^2\right\}^{1/2} \quad (8.4.21)$$

式中

$$\bar{V}_m=\frac{1}{M}\sum_{i=1}^M\bar{V}(X, \delta_i)$$

$$\bar{V}(X, \delta_i)=\frac{V(X, \delta_i)}{\omega}$$

$$\delta_i=\alpha_i-\alpha_0$$

式中, M 为工作行程区域内, 曲柄的工作压力角 α_p 的等分点数, α_i 为第 i 个等分点曲柄所在的位置角。

工作行程压力角 α_p 为

$$\alpha_p=\alpha_0+\pi-\arccos\frac{R^2+C_5^2-L^2}{2RC_5}-\arctan\frac{B_5}{A_5} \quad (8.4.22)$$

式中

$$A_5=a+e$$

$$B_5=b-\sqrt{l_1^2-e^2}$$

$$C_5=\sqrt{A_5^2+B_5^2}$$

同时, 为了降低滑块与导轨的磨损, 要求滑块与导轨的摩擦力在压力行程中所做的功尽量减少。在整个压力行程中, 冷挤压工艺的挤压力可近似地视为常数^[41], 设挤压机的公称压力为 P_g , 导轨与滑块的摩擦系数为 μ , 则导轨与滑块所损耗的摩擦功率为

$$A(X, \delta_i)=\mu P_g \tan \theta V(X, \delta_i) \quad (8.4.23)$$

为了减少计算量, 将一些常数项放到加权系数中去或将常数项视为 1。则使滑块与导轨的摩擦力在压力行程过程中所做的功尽量减少的目标函数可取为

$$f_2(X, \delta_i) = \sum_{i=1}^M \tan \theta \bar{V}(X, \delta_i) \quad (8.4.24)$$

J88 型冷挤压机拉力肘杆机构的优化设计是使滑块的挤压速度尽量均匀和使滑块与导轨的磨损最小。它是一个多目标函数的优化设计问题, 用线性加权法将多目标函数转化为单目标函数, 则 J88 型冷挤压机拉力肘杆机构优化设计的目标函数为

$$f(X) = W_1 f_1(X) + W_2 f_2(X) \quad (8.4.25)$$

式中

$$f_1(X) = \left\{ \sum_{i=1}^M [\bar{V}(X, \delta_i) - \bar{V}_m]^2 \right\}^{1/2}$$

$$f_2(X, \delta_i) = \sum_{i=1}^M \tan \theta \bar{V}(X, \delta_i)$$

W_1 和 W_2 为加权因子。

3. 约束条件

由于非独立构件参数的计算, 会使拉力肘杆机构中的四连杆机构自动地为曲柄摆杆机构, 故约束条件不需考虑四连杆机构成立的条件。考虑挤压机结构的布置、传动角的限制等, 可建立如下约束条件:

$$\begin{aligned} l_{1\min} &\leq l_1 \leq l_{1\max} \\ a_{\min} &\leq a \leq a_{\max} \\ e_{\min} &\leq e \leq e_{\max} \\ R_{\min} &\leq R \leq R_{\max} \\ b &\leq b_{\max} \\ \rho_b &\leq \rho_{\max} \\ \sqrt{l_3^2 - e^2} - l_1 - Y &\leq 0 \\ R_l - \sqrt{a^2 + (b - l_1)^2} &\leq 0 \\ \gamma_{\min} &\leq \arccos \frac{L^2 + l_1^2 - (\sqrt{a^2 + b^2} - R)^2}{2Ll_1} \\ \arccos \frac{L^2 + l_1^2 - (\sqrt{a^2 + b^2} + R)^2}{2Ll_1} &\leq \gamma_{\max} \end{aligned} \quad (8.4.26)$$

式中, R_l 为曲柄上大齿轮或球副蜗轮的半径; γ_{\min} 和 γ_{\max} 为 $\angle ABD$ 的最小值和最大值 (即 $\angle ABD$

的取值范围为 $[\gamma_{\min}, \gamma_{\max}]$); $l_{1\min}$ 和 $l_{1\max}$ 分别为 l_1 取值的上限和下限, 其余类推。

综上所述, J88 型冷挤压机拉力肘杆机构优化数学模型的标准形式为

$$\min f(X) = \min [W_1 f_1(X) + W_2 f_2(X)] \quad (X \in R^2) \quad (8.4.27)$$

满足于

$$\begin{aligned} g_1(X) &= l_{1\min} - l_1 \leq 0 \\ g_2(X) &= l_1 - l_{1\max} \leq 0 \\ g_3(X) &= a_{\min} - a \leq 0 \\ g_4(X) &= a - a_{\max} \leq 0 \\ g_5(X) &= e_{\min} - e \leq 0 \\ g_6(X) &= e - e_{\max} \leq 0 \\ g_7(X) &= R_{\min} - R \leq 0 \\ g_8(X) &= R - R_{\max} \leq 0 \\ g_9(X) &= b - b_{\max} \leq 0 \\ g_{10}(X) &= \rho_b - \rho_{\max} \leq 0 \\ g_{11}(X) &= \sqrt{l_3^2 - e^2} - l_1 - Y \leq 0 \\ g_{12}(X) &= R_t - \sqrt{a^2 + (b - l_1)^2} \leq 0 \\ g_{13}(X) &= \gamma_{\min} - \arccos \frac{L^2 + l_1^2 - (\sqrt{a^2 + b^2} - R)^2}{2Ll_1} \leq 0 \\ g_{14}(X) &= \arccos \frac{L^2 + l_1^2 - (\sqrt{a^2 + b^2} + R)^2}{2Ll_1} - \gamma_{\max} \leq 0 \end{aligned}$$

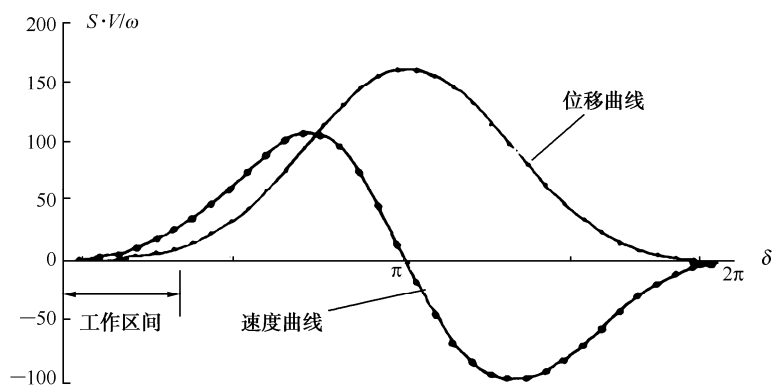
8.4.6 J88-400 型冷挤压机拉力肘杆机构优化设计

对 J88-400 型冷挤压机拉力肘杆机构进行优化设计, 取压机的技术参数为 $S_M = 160\text{mm}$, $S_p = 10\text{mm}$, $K = 1.05$, 原 J88-400 型冷挤压机拉力肘杆机构参数见文献[40]。

根据压机的具体情况, 取 $Y = 1300$, $260 \leq l_1 \leq 350$, $300 \leq a \leq 450$, $60 \leq e \leq 100$, $120 \leq R \leq 200$, $b_{\max} = 800$, $\rho_{\max} = 70^\circ$, $R_t = 500$, $\gamma_{\min} = 45^\circ$, $\gamma_{\max} = 135^\circ$, $M = 20$, $W_1 = 1$, $W_2 = 2$ 。

用复合形法求解, 其优化结果为: $a = 367.169$, $b = 651.308$, $l_1 = 311.895$, $l_3 = 1613.507$, $L = 661.783$, $R = 161.778$, $e = 71.889$, $f(X^*) = 14.88811$ 。

优化的拉力肘杆机构的滑块位移和速度曲线如图 8.4.5 所示, 由图 8.4.5 可知, 滑块在工作区间内的速度比较均匀。



8.5 JA88 型冷挤压机压力肘杆机构优化设计^[42]

8.5.1 JA88 型冷挤压机压力肘杆机构

JA88 型冷挤压机压力肘杆机构如图 8.5.1 所示, 主要部件有曲柄 R 、连杆 L 、摆杆 l_1 和压杆 l_2 等。与 J88 型冷挤压机拉力肘杆机构 (见 8.4 节) 一样, JA88 型冷挤压机压力肘杆机构也是由曲柄滑块机构加上曲柄摆杆机构组成。由于这两套机构均在接近下死点附近区域内工作, 这种挤压机滑块的工作行程速度缓慢, 有利于冷挤压工艺; 又由于机构具有急回特性, 对提高生产效率有利。

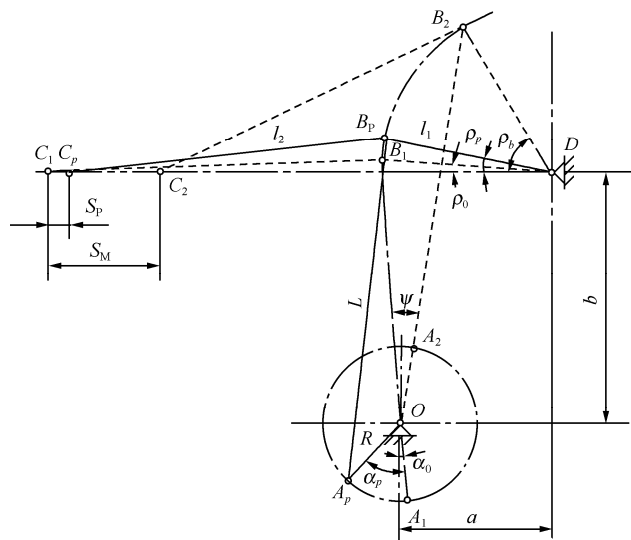


图 8.5.1 压力肘杆机构

图 8.5.1 中符号的意义分别是:

O ——曲柄回转中心;

D ——摆杆回转中心;

A ——曲柄铰链点;

B ——摆杆铰链点;

C ——压杆与滑块的铰链点;

R ——曲柄的长度;

L ——连杆的长度;

l_1 ——摆杆的长度;

l_2 ——压杆的长度;

a ——曲柄回转中心与摆杆回转中心的水平尺寸;

b ——曲柄回转中心与摆杆回转中心的垂直尺寸。

8.5.2 压力肘杆机构非独立构件尺寸参数的计算

压力肘杆机构由 l_1 、 l_2 、 R 、 L 、 a 和 b 等参数确定。为了精确地满足 JA88 型冷挤压机的滑块行程 S_M 、滑块的工作行程 S_p 和机构的急回特性系数 K 等技术参数的要求, 将机构的参数 l_1 、 l_2 和 a 定为独立参数, 而机构的参数 R 、 L 和 b 等定为非独立参数。

下面, 我们来讨论压力肘杆机构非独立参数的计算。

为了防止在一个循环中滑块出现两次下死点的现象, 应保证 $\rho_0 \geq 0^\circ$, ρ_0 为摆杆 l_1 与水平线的夹角 (见图 8.5.1)。考虑到构件尺寸的加工误差等, 一般取 $\rho_0 = 0.5^\circ \sim 1.5^\circ$ 。

设 ρ_b 为摆杆 l_1 与水平线的最大夹角, 由图 8.5.1 中的几何关系, 可得滑块行程

$$S_M = l_1 \cos \rho_0 + \sqrt{l_2^2 - l_1^2 \sin^2 \rho_0} - l_1 \cos \rho_b - \sqrt{l_2^2 - l_1^2 \sin^2 \rho_b}$$

解上式得

$$\rho_b = \arccos \frac{l_2^2 - l_1^2 - A_1^2}{2A_1 l_1} \quad (8.5.1)$$

式中, $A_1 = S_M - l_1 \cos \rho_0 - \sqrt{l_2^2 - l_1^2 \sin^2 \rho_0}$ 。

同理可得

$$\rho_p = \arccos \frac{l_2^2 - l_1^2 - A_2^2}{2A_2 l_1} \quad (8.5.2)$$

式中, $A_2 = S_p - l_1 \cos \rho_0 - \sqrt{l_2^2 - l_1^2 \sin^2 \rho_0}$; ρ_p 为开始压力行程时摆杆 l_1 与水平线的夹角; S_p 为滑块的工作行程。

设 ψ 为曲柄和连杆重合时, 曲柄和连杆两个极限位置的夹角, 它与机构的急回特性系数 K 的关系为

$$\psi = 180^\circ \frac{K-1}{K+1} \quad (8.5.3)$$

式中, $K = \frac{V_2}{V_1}$, V_1 、 V_2 分别为滑块回程和滑块向前运行的平均速度。

由图 8.5.1 中的几何关系, 可得

$$\psi = \arctan \frac{l_1 \cos \rho_0 - a}{b + l_1 \sin \rho_0} - \arctan \frac{l_1 \cos \rho_b - a}{b + l_1 \sin \rho_b}$$

解上式, 可得

$$b = \frac{-B_3 + \sqrt{B_3^2 - 4A_3C_3}}{2A_3} \quad (8.5.4)$$

式中

$$A_3 = \tan \psi$$

$$B_3 = l_1 [A_3 (\sin \rho_0 + \sin \rho_b) + \cos \rho_b - \cos \rho_0]$$

$$C_3 = A_3 [l_1^2 \sin \rho_0 \sin \rho_b + (l_1 \cos \rho_0 - a)(l_1 \cos \rho_b - a)] + l_1 \sin \rho_0 (l_1 \cos \rho_b - a) - l_1 \sin \rho_b (l_1 \cos \rho_0 - a)$$

由图 8.5.1 中的几何关系, 还可得

$$L + R = \sqrt{(l_1 \cos \rho_b - a)^2 + (b + l_1 \sin \rho_b)^2}$$

$$L - R = \sqrt{(l_1 \cos \rho_0 - a)^2 + (b + l_1 \sin \rho_0)^2}$$

求解上面两式, 可得连杆长度 L 和曲柄半径 R 的计算公式为

$$L = \frac{1}{2} \left[\sqrt{(l_1 \cos \rho_0 - a)^2 + (b + l_1 \sin \rho_0)^2} + \sqrt{(l_1 \cos \rho_b - a)^2 + (b + l_1 \sin \rho_b)^2} \right] \quad (8.5.5)$$

$$R = L - \sqrt{(l_1 \cos \rho_0 - a)^2 + (b + l_1 \sin \rho_0)^2} \quad (8.5.6)$$

8.5.3 压力肘杆机构的运动分析

因为摆杆 l_1 与水平线的夹角 ρ_0 作为滑块的下死点, 取滑块下死点 C_1 作为滑块位移 S 的计算起点, 由图 8.5.2 中的几何关系可得滑块的位移为

$$S = l_1 \cos \rho_0 + \sqrt{l_2^2 - l_1^2 \sin^2 \rho_0} - l_1 \cos \rho - l_2 \cos \theta \quad (8.5.7)$$

式中

$$\rho = \arctan \frac{A_4}{B_4} + \arccos \frac{C_4^2 + l_1^2 - L^2}{2C_4 l_1} - 90^\circ = \arctan \frac{A_4}{B_4} - \arcsin \frac{C_4^2 + l_1^2 - L^2}{2C_4 l_1}$$

$$\theta = \arcsin \frac{l_1 \sin \rho}{l_2}$$

$$A_4 = a + R \sin \delta$$

$$B_4 = b + R \cos \delta$$

$$C_4 = \sqrt{A_4^2 + B_4^2}$$

$$\delta = \alpha - \alpha_0$$

$$\alpha_0 = \arctan \frac{l_1 \cos \rho_0 - a}{b + l_1 \sin \rho_0}$$

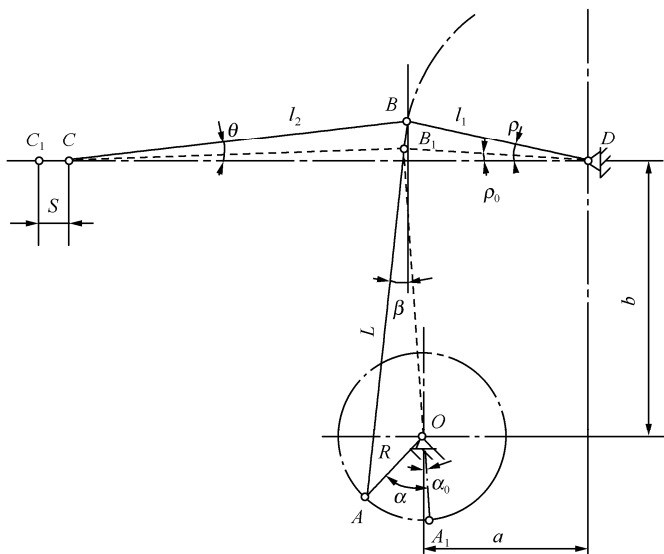


图 8.5.2 压力肘杆机构运动分析图

将式 (8.5.7) 对时间 t 求导, 整理可得滑块的速度为

$$V = \frac{dS}{dt} = \frac{R\omega \sin(\rho + \theta) \sin(\delta - \beta)}{\cos \theta \cos(\rho - \beta)} \quad (8.5.8)$$

其中, $\beta = \arctan \frac{R \sin \delta + a - l_1 \cos \rho}{R \cos \delta + b + l_1 \sin \rho}$, $\omega = \frac{d\alpha}{dt}$ 为曲柄的角速度。

8.5.4 压力肘杆机构的受力分析与曲柄轴摩擦扭矩的计算

设滑块上的作用力为压机的公称压力 P_g , 由图 8.5.3 和力的三角形关系, 可得

$$P_{BC} = \frac{P_g}{\cos \theta} \quad (8.5.9)$$

$$P_N = P_g \tan \theta \quad (8.5.10)$$

$$P_{AB} = \frac{P_g \sin(\rho + \theta)}{\cos \theta \cos(\rho - \beta)} \quad (8.5.11)$$

$$P_{BD} = \frac{P_g \cos(\theta + \beta)}{\cos \theta \cos(\rho - \beta)} \quad (8.5.12)$$

若考虑摩擦的影响, 设摩擦系数为 μ , 则侧压力为

$$\bar{P}_N = \frac{P_g \cos \varphi \sin(\theta + \lambda)}{\cos(\varphi + \theta + \lambda)} \quad (8.5.13)$$

式中, $\varphi = \arctan \mu$, $\lambda = \arcsin \frac{\mu(R_B + R_C)}{l_2}$, R_B 为销轴 B 的半径, R_C 为销轴 C 的半径。

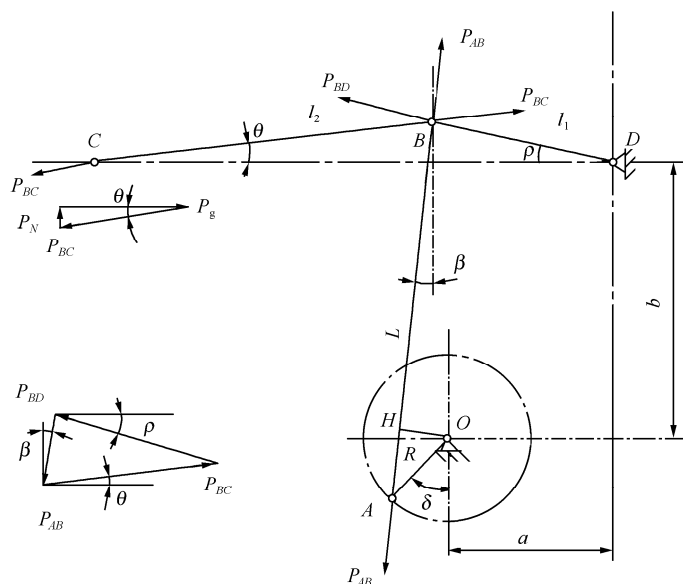


图 8.5.3 压力肘杆机构受力分析图

挤压时曲柄轴所需要的扭矩 M_q , 除使工件变形所需要的扭矩 M_l 外, 还要克服由于各运动部分的摩擦而引起的摩擦力矩 M_μ , 故有

$$M_q = M_l + M_\mu \quad (8.5.14)$$

由图 8.5.3, 可得工件变形所需要的扭矩为

$$M_l = P_{AB} \overline{OH} = \frac{R P_g \sin(\rho + \theta) \sin(\delta - \beta)}{\cos \theta \cos(\rho - \beta)} \quad (8.5.15)$$

由功能平衡原理知, 曲柄轴所增加的摩擦力矩 M_μ 在单位时间里所做的功 (即功率) 等于单位时间内各处摩擦所损耗的功, 故有

$$M_\mu \omega = A_N + A_O + A_A + A_B + A_C + A_D \quad (8.5.16)$$

式中, ω 为曲柄轴的角速度。

设摩擦系数为 μ , 则滑块与导轨所损耗的摩擦功率为

$$A_N = \mu P_N V = \mu P_g \tan \theta V \quad (8.5.17)$$

由于在工作行程中, 滑块的速度 V 和 θ 角都很小, 故 A_N 可忽略不计。

设曲轴两支承的半径为 R_O , 则曲轴两支承处摩擦所损耗的功率 A_O 为

$$A_O = \mu R_O P_{AB} \omega = \frac{\mu R_O \omega P_g \sin(\rho + \theta)}{\cos \theta \cos(\rho - \beta)} \quad (8.5.18)$$

设曲柄颈的半径为 R_A , 则曲柄颈 A 处摩擦所损耗的功率为

$$A_A = \mu R_A P_{AB} \left(\omega - \frac{d\beta}{dt} \right) = \frac{\mu R_A \omega P_g \sin(\rho + \theta)}{\cos \theta \cos(\rho - \beta)} \left[1 - \frac{R \cos(\delta - \rho)}{L \cos(\rho - \beta)} \right] \quad (8.5.19)$$

设销轴 D 的半径为 R_D , 则销轴 D 处摩擦所损耗的功率为

$$A_D = \mu R_D P_{BD} \frac{d\rho}{dt} = \frac{\mu R_D R \omega P_g \cos(\beta + \theta) \sin(\delta - \beta)}{l_1 \cos \theta \cos^2(\rho - \beta)} \quad (8.5.20)$$

设销轴 C 的半径为 R_C , 则销轴 C 处摩擦所损耗的功率为

$$A_C = \mu R_C P_{BC} \frac{d\theta}{dt} = \frac{\mu R_C R \omega P_g \cos \rho \sin(\delta - \beta)}{l_2 \cos^2 \theta \cos(\rho - \beta)} \quad (8.5.21)$$

设销轴 B 的半径为 R_B , 则销轴 B 处摩擦所损耗的功率为

$$\begin{aligned} A_B &= \mu R_B \left[-P_{BC} \frac{d(180^\circ - \theta - \rho)}{dt} + \left| P_{AB} \frac{d(90^\circ - \rho + \beta)}{dt} \right| \right] \\ &= \frac{\mu R_B R \omega P_g}{\cos \theta \cos(\rho - \beta)} \left\{ \sin(\delta - \beta) \left[\frac{\cos \rho}{l_2 \cos \theta} + \frac{1}{l_1} \right] \right. \\ &\quad \left. + \frac{\sin(\rho + \theta)}{\cos(\rho - \beta)} \left| \frac{\sin(\delta - \beta)}{l_1} - \frac{\cos(\delta - \rho)}{L} \right| \right\} \quad (8.5.22) \end{aligned}$$

注: 上式中的负号和取绝对值的理由分别是在工作行程阶段, 由于 $(180^\circ - \theta - \rho)$ 角随 α 的增加而减少; $(90^\circ - \rho - \beta)$ 角随 α 的增加有时增加, 有时减少; 而摩擦总是消耗有用功。

将式 (8.5.18) ~ 式 (8.5.22) 代入式 (8.5.16), 整理可得曲柄轴所增加的摩擦力矩 M_μ 为

$$\begin{aligned} M_\mu &= \frac{\mu R P_g}{\cos \theta \cos(\rho - \beta)} \left\{ \frac{(R_O + R_A) \sin(\rho + \theta)}{R} \right. \\ &\quad \left. + \frac{\sin(\rho + \theta)}{\cos(\rho - \beta)} \left[\left| \frac{R_B \sin(\delta - \beta)}{l_1} - \frac{R_B \cos(\delta - \rho)}{L} \right| - \frac{R_A \cos(\delta - \rho)}{L} \right] \right\} \end{aligned}$$

$$+\sin(\delta-\beta)\left[\frac{R_D\cos(\beta+\theta)}{l_1\cos(\rho-\beta)}+\frac{(R_C+R_B)\cos\rho}{l_2\cos\theta}+\frac{R_B}{l_1}\right]\quad (8.5.23)$$

8.5.5 压力肘杆机构优化设计数学模型的建立

1. 设计变量

在 8.5.2 节中, 压力肘杆机构构件参数 l_1 、 l_2 、 R 、 L 、 a 和 b 分为独立参数和非独立参数。非独立参数由独立参数和冷挤压机的滑块行程 S_M 、滑块的工作行程 S_P 和机构的急回特性系数 K 等技术参数确定。压力肘杆机构的优化设计的设计变量由机构的独立参数组成, 因此, 压力肘杆机构的设计变量为

$$X=(x_1, x_2, x_3)^T=(l_1, l_2, a)^T \quad (8.5.24)$$

2. 目标函数

挤压工艺要求滑块的挤压速度尽量均匀, 一般认为较为理想的挤压速度为 $100\sim 400\text{mm/s}$ 。同时, 为了降低飞轮的转动惯量和节约能源, 要求在挤压过程中消耗的能量最小; 为了减少滑块对导轨的摩擦, 要求滑块对导轨的侧压力尽量小。分别用函数 $f_1(X)$ 来表示滑块的挤压速度的均匀程度、用函数 $f_2(X)$ 来表示在挤压过程中消耗的能量大小、用 $f_3(X)$ 来表示滑块对导轨的侧压力的大小。将某些常数放到加权因子 W_1 、 W_2 和 W_3 中或省略, 则可建立压力肘杆机构构件优化设计的目标函数如下:

$$f(X)=W_1f_1(X)+W_2f_2(X)+W_3f_3(X) \quad (8.5.25)$$

式中

$$f_1(X)=\left\{\sum_{i=1}^M[\bar{V}(X, \alpha_i)-\bar{V}_m]^2\right\}^{1/2}$$

$$f_2(X)=\sum_{i=1}^M\frac{M_q(X, \alpha_i)}{P_g}$$

$$f_3(X)=\max\left[\frac{\bar{P}_N(X, \alpha_i)}{P_g}\right]=\frac{\bar{P}_N(X, \alpha_p)}{P_g}=\frac{\cos\varphi\sin(\theta+\lambda)}{\cos(\varphi+\theta+\lambda)}$$

这里

$$\bar{V}_m=\frac{1}{M}\sum_{i=1}^M\bar{V}(X, \alpha_i)$$

而

$$\bar{V}(X, \alpha_i)=\frac{V(X, \alpha_i)}{\omega}$$

其中, M 为挤压工作行程区域内曲柄的工作压力角 α_p 的等分点数; α_i 为第 i 个等分点曲

柄所在的位置角。

3. 约束条件

由于非独立构件参数的计算, 会使压力肘杆机构中的四杆机构自动地为曲柄摆杆机构, 故约束条件不需考虑四杆机构成立的条件。考虑到压机结构的布置、传动角的限制等, 可建立如下约束条件:

$$\begin{aligned}
 l_{1\min} &\leq l_1 \leq l_{1\max} \\
 a_{\min} &\leq a \leq a_{\max} \\
 l_2 &\geq l_1 \\
 l_1 + l_2 &\leq [l_1 + l_2]_{\max} \\
 \rho_b &\leq \rho_{\max} \\
 b_{\min} &\leq b \leq b_{\max} \\
 B_3^2 - 4A_3C_3 &\geq 0 \\
 \arccos \frac{L^2 + l_1^2 - (\sqrt{a^2 + b^2} + R)^2}{2Ll_1} &\leq \gamma_{\max} \\
 \arccos \frac{L^2 + l_1^2 - (\sqrt{a^2 + b^2} - R)^2}{2Ll_1} &\geq \gamma_{\min}
 \end{aligned} \tag{8.5.26}$$

式中, γ_{\min} 和 γ_{\max} 为 $\angle ABD$ 的最小值和最大值 (即 $\angle ABD$ 的取值范围为 $[\gamma_{\min}, \gamma_{\max}]$); $l_{1\min}$ 和 $l_{1\max}$ 分别为 l_1 取值的上下限, 其余类推。

综上所述, JA88 型冷挤压机压力肘杆机构优化数学模型的标准形式为

$$\min f(X) = \min \{W_1 f_1(X) + W_2 f_2(X) + W_3 f_3(X)\} \quad (X \in R^3) \tag{8.5.27}$$

满足于

$$\begin{aligned}
 g_1(X) &= l_{1\min} - l_1 \leq 0 \\
 g_2(X) &= l_1 - l_{1\max} \leq 0 \\
 g_3(X) &= a_{\min} - a \leq 0 \\
 g_4(X) &= a - a_{\max} \leq 0 \\
 g_5(X) &= l_1 - l_2 \leq 0 \\
 g_6(X) &= l_1 + l_2 - [l_1 + l_2]_{\max} \leq 0 \\
 g_7(X) &= \rho_b - \rho_{\max} \leq 0 \\
 g_8(X) &= b_{\min} - b \leq 0 \\
 g_9(X) &= b - b_{\max} \leq 0
 \end{aligned}$$

$$g_{10}(X)=4A_3C_3-B_3^2\leq 0$$

$$g_{11}(X)=\arccos \frac{L^2+l_1^2-(\sqrt{a^2+b^2}+R)^2}{2Ll_1}-\gamma_{\max}\leq 0$$

$$g_{12}(X)=\gamma_{\min}-\arccos \frac{L^2+l_1^2-(\sqrt{a^2+b^2}-R)^2}{2Ll_1}\leq 0$$

8.5.6 JA88-160 型冷挤压机压力肘杆机构优化设计

对 JA88-160 型冷挤压机压力肘杆机构进行优化设计, 取冷挤压机的技术参数为 $S_M=250\text{mm}$, $S_p=5\text{mm}$, $K=1.2$ 。

根据压机的具体情况, 取 $R_O=65\text{mm}$, $R_A=65\text{mm}$, $R_B=40\text{mm}$, $R_C=30\text{mm}$, $R_D=30\text{mm}$, $200\leq l_1\leq 350$, $150\leq a\leq 350$, $470\leq b\leq 600$, $\rho_{\max}=70^\circ$, $\gamma_{\min}=30^\circ$, $\gamma_{\max}=120^\circ$, $120\leq R\leq 200$, $b_{\max}=800\text{mm}$, $M=12$, $\mu=0.06$, $W_1=5$, $W_2=1$, $W_3=1500$, $[l_1+l_2]_{\max}=1050\text{mm}$ 。

用复合形法求解 JA88-160 型冷挤压机压力肘杆机构优化设计问题, 其优化结果为: $l_1=297.947\text{mm}$, $l_2=750.548\text{mm}$, $a=262.196\text{mm}$, $b=470.002\text{mm}$, $L=621.088\text{mm}$, $R=145.742\text{mm}$, $f(X^*)=304.0292$, $f_1(X^*)=22.220\ 14$, $f_2(X^*)=91.484\ 62$, $\bar{P}_N(X^*, \alpha_p)=10.82(T)$ 。

优化的压力肘杆机构在工作压力行程区间内有关计算数据如表 8.5.1 所示。

表 8.5.1 优化机构有关计算数据

| $\alpha(^{\circ})$ | $\rho(^{\circ})$ | $S(\text{mm})$ | V/ω | M_q/P_g |
|--------------------|------------------|----------------|------------|-----------|
| 0 | 0.769230 | -0.000 | 0.0000 | 0.1403 |
| 4.611194 | 0.838853 | 0.007 | 0.1838 | 0.5122 |
| 9.222387 | 1.047215 | 0.032 | 0.4577 | 0.9977 |
| 13.833851 | 1.393470 | 0.086 | 0.9106 | 1.6847 |
| 18.444774 | 1.876562 | 0.186 | 1.6288 | 2.6582 |
| 23.055968 | 2.495210 | 0.357 | 2.6947 | 3.9994 |
| 27.667161 | 3.247704 | 0.631 | 4.1854 | 5.7840 |
| 32.278355 | 4.132027 | 1.044 | 6.1711 | 8.0803 |
| 36.889548 | 5.145612 | 1.639 | 8.7128 | 10.9632 |
| 41.500742 | 6.285525 | 2.463 | 11.8607 | 14.4787 |
| 46.111935 | 7.548215 | 3.565 | 15.6521 | 18.6593 |
| 50.723129 | 8.929742 | 5.000 | 20.1099 | 23.5268 |

8.6 八连杆双动拉延压力机内外滑块执行机构优化设计^[43~45]

8.6.1 板料拉延工艺

利用具有一定圆角半径的拉延模(亦称为拉深模), 将平板毛坯或开口空心毛坯冲压成容器状零件的冲压过程称为拉延。

拉延工艺是主要的冲压工序之一，其应用很广，如汽车和拖拉机的一些零件、覆盖件，仪器仪表的壳体件，以及众多的日用品等都是应用拉延工艺成型的。

对于深拉延件，变形程度较大，为了防止起皱，需要对毛坯施加压边力，如图 8.6.1 所示。采用双动拉延压力机进行深拉延件加工时，压边力是由双动拉延压力机的外滑块施加的，而拉延力是由压机的内滑块施加的，如图 8.6.2 所示^[29]。

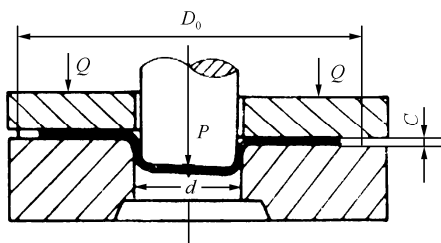


图 8.6.1 带压边圈的冲压模具

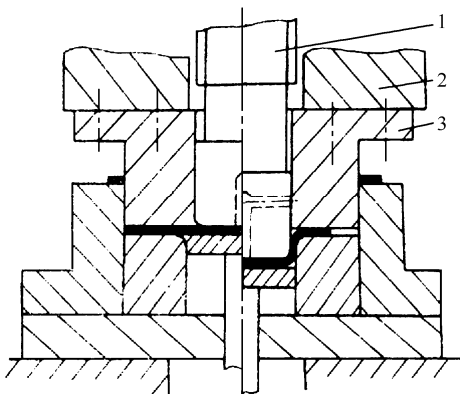
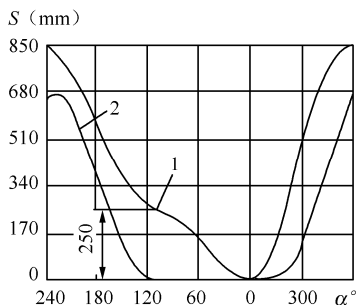


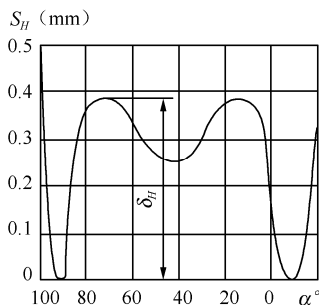
图 8.6.2 双动拉延压力机与模具的关系

1—内滑块 2—外滑块 3—压边圈

为了保证拉伸工艺顺利进行，首先，要保证外滑块带动模具的压边圈将坯料的外缘紧紧压住。在内滑块拉伸过程中，压边圈静止不动，拉延工艺允许外滑块位移的波动量仅为 $0.03 \sim 0.05\text{mm}$ ^[41]。其次，内滑块和外滑块的运动应保持一定的关系，即内滑块开始拉伸之前，外滑块提前压紧坯料，一般提前角为 15° 左右。再其次，为了保证拉伸结束时拉伸零件顶在下模平面上，并防止在内滑块回程过程中上模将拉伸零件带上，使零件破坏或产生事故，所以，外滑块的回程应稍滞后于内滑块，一般滞后角为 10° 左右。最后，在内滑块在拉伸过程中，要求滑块的速度缓慢，并且均匀。图 8.6.3 (a) 为原苏联生产的 6.3/4MN 板冲压力机内、外滑块位移曲线图^[46]。



(a) 内滑块位移曲线 (1) 和外滑块位移曲线 (2)



(b) 外滑块在停留阶段的位移曲线

图 8.6.3 6.3/4MN 压力机的滑块位移曲线图

8.6.2 八连杆双动拉延压力机内外滑块执行机构

八连杆双动拉延压力机内、外滑块执行机构如图 8.6.4 所示, 它们具有公共的四杆机构 $OACD$ 。内、外滑块执行机构的曲柄为 OA , 其尺寸参数为 R ; 其他构件以及构件尺寸参数如图 8.6.4 所示。通过仔细地分析发现, 外滑块机构只是内滑块机构的一种特殊形式, 外滑块机构是内滑块机构构件 OB 的尺寸参数 $R_1=0$ 和偏置 $e=0$ 的一种特殊形式。因此, 在下面推导机构的运动参数公式时, 我们只需要推导内滑块机构的运动参数公式即可。由于这种机构复杂, 因此, 用传统的作图方法来确定内、外滑块执行机构构件尺寸参数, 不但不能使压力机达到最优状态, 也无法使压力机精确满足预先规定的运动技术参数。特别是在压紧角内, 外滑块位移的波动量大, 达不到拉延工艺压边允许外滑块位移的波动量为 $0.03\sim 0.05\text{mm}$ 的要求^[41], 从而严重影响压力机的工作性能。从图 8.6.3 (b) 中可以清楚地看出, 原苏联生产的 6.3/4MN 板冲压力机在压边阶段, 外滑块位移的波动量约为 0.43mm , 远远大于拉延工艺要求的外滑块的波动量 $0.03\sim 0.05\text{mm}$, 从而严重影响压力机的工作性能。

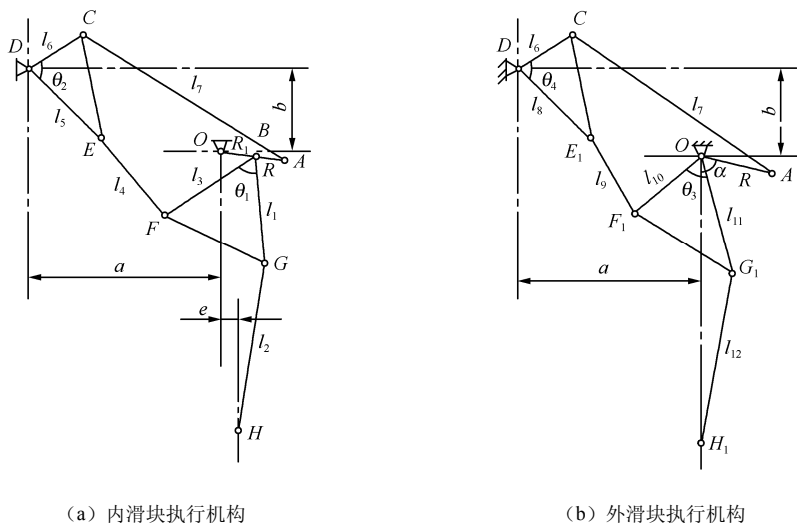


图 8.6.4 八连杆双动拉延压力机内外滑块执行机构

8.6.3 机构构件非独立构件尺寸参数的计算

为了使压力机内滑块行程 S_n 、内滑块的压力行程 S_g 、外滑块行程 S_m 、在压紧角 α_p 内外滑块的波动量 ΔS 精确地为预先给定的值, 将机构尺寸参数 l_1 、 l_2 、 l_3 、 l_5 、 l_6 、 l_8 、 l_{10} 、 l_{11} 、 l_{12} 、 a 、 b 、 e 、 θ_1 、 θ_2 、 θ_3 和 θ_4 定为独立参数, 而其他参数 l_4 、 l_7 、 l_9 、 R 和 R_1 定为非独立参数。独立参数在优化设计中作为设计变量, 因此, 下面我们仅讨论非独立构件尺寸参数的计算和处理方法。

将构件 l_{11} 与 l_{12} 成直线时作为外滑块的下死点。设 ρ_b 为构件 l_{11} 与铅垂线的最大夹角，由图 8.6.5 中的几何关系，可得外滑块的行程为

$$S_m = l_{11} + l_{12} - l_{11} \cos \rho_b - \sqrt{l_{12}^2 - l_{11}^2 \sin^2 \rho_b} \quad (8.6.1)$$

解上式得

$$\rho_b = \arccos \frac{l_{12}^2 - l_{11}^2 - A_1^2}{2A_1 l_{11}} \quad (8.6.2)$$

式中， $A_1 = S_m - l_{11} - l_{12}$ 。

同理可得

$$\rho_0 = \arccos \frac{l_{12}^2 - l_{11}^2 - A_2^2}{2A_2 l_{11}} \quad (8.6.3)$$

式中， $A_2 = \Delta S - l_{11} - l_{12}$ ， R_0 为在压紧角内，允许外滑块位移的波动量为 ΔS 时，构件 l_{11} 与铅垂线左右摆动的最大夹角，见图 8.6.5。

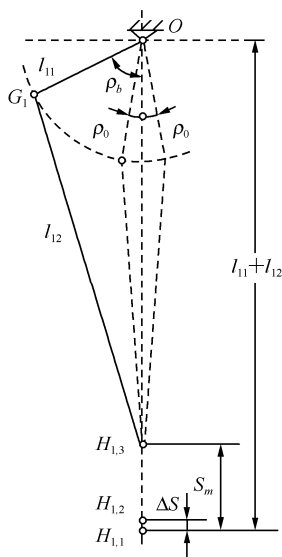


图 8.6.5 滑块运动分析图 (1)

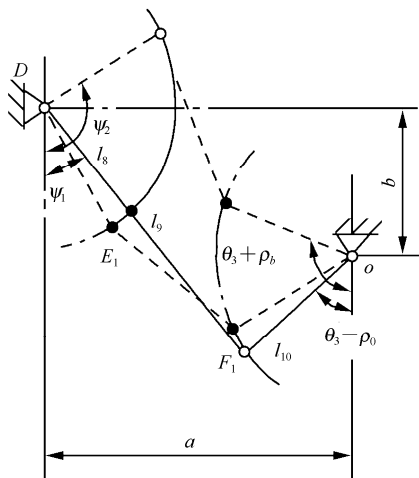


图 8.6.6 运动分析图 (2)

当构件 l_8 和 l_9 在一条直线上时，如图 8.6.6 所示，由几何关系得

$$l_9 = \sqrt{[a - l_{10} \sin(\theta_3 - \rho_0)]^2 + [b + l_{10} \cos(\theta_3 - \rho_0)]^2} - l_8 \quad (8.6.4)$$

由图 8.6.7 中的几何关系可得连杆长度 l_7 和曲柄半径 R 分别为：

$$l_7 = \frac{1}{2} \left\{ \sqrt{[a - l_6 \sin(\psi_1 + \theta_4)]^2 + [b - l_6 \cos(\psi_1 + \theta_4)]^2} + \right.$$

$$\sqrt{[a - l_6 \sin(\psi_2 + \theta_4)]^2 + [b - l_6 \cos(\psi_2 + \theta_4)]^2} \quad (8.6.5)$$

$$R = l_7 - \sqrt{[a - l_6 \sin(\psi_1 + \theta_4)]^2 + [b - l_6 \cos(\psi_1 + \theta_4)]^2} \quad (8.6.6)$$

式中

$$\psi_1 = \arctan \frac{A_3}{B_3} - \arccos \frac{l_8^2 + C_3^2 - l_9^2}{2l_8 C_3}$$

$$\psi_2 = \begin{cases} \arctan \frac{A_4}{B_4} + \arccos \frac{l_8^2 + C_4^2 - l_9^2}{2l_8 C_4} & \frac{A_4}{B_4} \geq 0 \\ \pi + \arctan \frac{A_4}{B_4} + \arccos \frac{l_8^2 + C_4^2 - l_9^2}{2l_8 C_4} & \frac{A_4}{B_4} < 0 \end{cases}$$

$$A_3 = a - l_{10} \sin(\theta_3 + \rho_0)$$

$$B_3 = b + l_{10} \cos(\theta_3 + \rho_0)$$

$$C_3 = \sqrt{A_3^2 + B_3^2}$$

$$A_4 = a - l_{10} \sin(\theta_3 + \rho_b)$$

$$B_4 = b + l_{10} \cos(\theta_3 + \rho_b)$$

$$C_4 = \sqrt{A_4^2 + B_4^2}$$

为了改善内滑块的受力状态, 取构件 R_1 、 l_1 和 l_2 成一条直线作为内滑块位移的下死点, 由几何关系可得

$$l_4 = \sqrt{A_6^2 + B_6^2} \quad (8.6.7)$$

式中

$$A_6 = a + R_1 \sin \alpha_1 - l_3 \sin(\theta_1 - \alpha_1) - l_5 \sin \psi_3$$

$$B_6 = b + R_1 \cos \alpha_1 - l_3 \cos(\theta_1 - \alpha_1) - l_5 \cos \psi_3$$

其中

$$\alpha_1 = \arcsin \frac{e}{R_1 + l_1 + l_2}$$

$$\psi_3 = \arctan \frac{A_5}{B_5} - \arccos \frac{l_6^2 + C_5^2 - l_7^2}{2l_6 C_5} - \theta_2$$

而

$$A_5 = a + R \sin \alpha_1$$

$$B_5 = b + R \cos \alpha_1$$

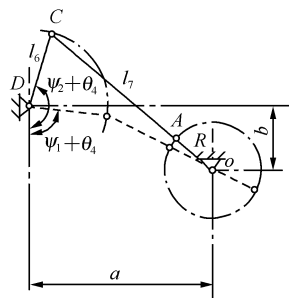


图 8.6.7 运动分析图 (3)

$$C_5 = \sqrt{A_5^2 + B_5^2}$$

用改变 R_1 的方法来改变内滑块的行程 S_n , 使内滑块的行程 S_n 精确地为预先给定的值。可以证明内滑块行程 S_n 是 R_1 的单调函数, 可用一维优化方法来确定 R_1 。对于小于内滑块的压力行程 S_g 的机构, 对优化设计的目标函数进行惩罚, 使 S_g 精确地为预先规定值。

8.6.4 内、外滑块位移和速度公式的推导

由于 l_4 是在构件 R_1 、 l_1 和 l_2 成一条直线的条件下计算出来的, 构件 R_1 、 l_1 和 l_2 成一条直线时, 内滑块所在的位置是内滑块下死点, 因此, 内滑块的位移为

$$S = \sqrt{(R_1 + l_1 + l_2)^2 - e^2} - R_1 \cos \alpha - l_1 \cos \beta_1 - l_2 \cos \beta_2 \quad (8.6.8)$$

式中

$$\beta_1 = \rho_2 - \rho_1 + \theta_1 - \frac{\pi}{2}$$

$$\beta_2 = \arcsin \frac{R_1 \sin \alpha + l_1 \sin \beta_1 - e}{l_2}$$

其中

$$\rho_1 = \arctan \frac{B_7}{A_7}$$

$$\rho_2 = \arccos \frac{l_3^2 + C_7^2 - l_4^2}{2l_3 C_7}$$

而

$$A_7 = a + R_1 \sin \alpha - l_5 \sin \psi$$

$$B_7 = b + R_1 \cos \alpha - l_5 \cos \psi$$

$$C_7 = \sqrt{A_7^2 + B_7^2}$$

$$\psi = \psi_4 + \psi_5 - \theta_2$$

这里

$$\psi_4 = \begin{cases} \arctan \frac{A_8}{B_8} & B_8 \geq 0 \\ \pi + \arctan \frac{A_8}{B_8} & B_8 < 0 \end{cases}$$

$$\psi_5 = \arccos \frac{l_6^2 + C_8^2 - l_7^2}{2l_6 C_8}$$

而

$$A_8 = a + R \sin \alpha$$

$$B_8 = b + R \cos \alpha$$

$$C_8 = \sqrt{A_8^2 + B_8^2}$$

式中, ρ_1 、 ρ_2 和 ψ_4 等角度的意义如图 8.6.8 所示。

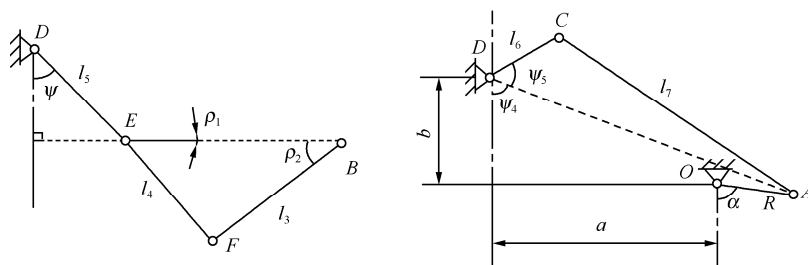


图 8.6.8 滑块运动参数图

内滑块位移对时间求导可得内滑块的运动速度 V , 由式 (8.6.8) 得

$$V = \frac{dS}{dt} = R_1 \omega \sin \alpha + l_1 \sin \beta_1 \frac{d\beta_1}{dt} + l_2 \sin \beta_2 \frac{d\beta_2}{dt} \quad (8.6.9)$$

式中

$$\frac{d\beta_1}{dt} = \frac{d\rho_2}{dt} - \frac{d\rho_1}{dt}$$

$$\frac{d\beta_2}{dt} = \frac{R_1 \omega \cos \alpha + l_1 \cos \beta_1 \frac{d\beta_1}{dt}}{l_2 \cos \beta_2}$$

其中

$$\frac{d\rho_1}{dt} = \frac{A_7(l_5 \sin \psi \frac{d\psi}{dt} - R_1 \omega \sin \alpha) + B_7(l_5 \cos \psi \frac{d\psi}{dt} - R_1 \omega \cos \alpha)}{C_7^2}$$

$$\frac{d\rho_2}{dt} = \frac{l_3^2 - C_7^2 - l_4^2}{2l_3 C_7^2 \sin \rho_2} \frac{dC_7}{dt}$$

而

$$\frac{dC_7}{dt} = \frac{A_7(R_1 \omega \cos \alpha - l_5 \cos \psi \frac{d\psi}{dt}) + B_7(l_5 \sin \psi \frac{d\psi}{dt} - R_1 \omega \sin \alpha)}{C_7}$$

这里

$$\frac{d\psi}{dt} = \frac{d\psi_4}{dt} + \frac{d\psi_5}{dt}$$

$$\frac{d\psi_4}{dt} = \frac{R\omega(B_8 \cos \alpha + A_8 \sin \alpha)}{C_8^2}$$

$$\frac{d\psi_5}{dt} = \frac{R\omega(C_8^2 - l_6^2 + l_7^2)(B_8 \sin \alpha - A_8 \cos \alpha)}{2l_6 C_8^3 \sin \psi_5}$$

$$\omega = \frac{d\alpha}{dt}$$

式中, ω 为曲柄的角速度。

由图 8.6.4 可知, 外滑块执行机构是内滑块执行机构 $e=0$ 和 $R_1=0$ 的一种特殊机构形式。内滑块的位移计算公式 (8.6.8) 和速度计算公式 (8.6.9) 也适应于外滑块, 即在式 (8.6.8) 和式 (8.6.9) 中令 $e=0$ 和 $R_1=0$ 即可得到外滑块的位移计算公式和速度计算公式。

8.6.5 八连杆内、外滑块执行机构优化设计目标函数的建立

1. 设计变量

在 8.6.3 节中, 将机构尺寸参数 l_1 、 l_2 、 l_3 、 l_5 、 l_6 、 l_8 、 l_{10} 、 l_{11} 、 l_{12} 、 a 、 b 、 e 、 θ_1 、 θ_2 、 θ_3 和 θ_4 定为独立参数, 而其他机构尺寸参数 l_4 、 l_7 、 l_9 、 R 和 R_1 定为非独立参数。非独立构件尺寸参数由压力机内滑块行程 S_n 、内滑块的压力行程 S_g 、外滑块行程 S_m 、在压紧角 σ_p 内外滑块的波动量 ΔS 精确地为预先给定的值, 以及独立构件尺寸参数来确定。八连杆双动拉延压力机内、外滑块执行机构优化设计的设计变量由机构的独立参数组成, 因此, 八连杆内、外滑块执行机构优化设计变量为

$$X = (x_1, x_2, \dots, x_{16})^T = (l_1, l_2, l_3, l_5, l_6, l_8, l_{10}, l_{11}, l_{12}, a, b, e, \theta_1, \theta_2, \theta_3, \theta_4)^T \quad (8.6.10)$$

2. 目标函数

为了出色地完成拉伸工作, 拉伸滑块在工作区间的速度应为 300~500mm/s (低碳薄钢板), 130~180mm/s (抗腐蚀钢板), 500~660mm/s (铝和青铜), 80~220mm/s (锌) [46]。内滑块在压力行程阶段速度均匀和较低的速度有利于拉延成形工艺, 因此, 要求内滑块在压力行程阶段速度尽量均匀, 并且平均速度尽量低。同时, 使内滑块的压力行程 S_g 满足预先给定的值 S_{g1} 。分别用函数 $f_1(X)$ 来表示在压力行程阶段中内滑块速度的均匀程度、用函数 $f_2(X)$ 来表示在压力行程阶段中内滑块平均速度的高低、用 $f_3(X)$ 来表示内滑块的压力行程 S_g 满足预先给定的值 S_{g1} 之差。八连杆双动拉延压力机内、外滑块执行机构优化设计是一个多目标函数的优化设计问题, 用线性加权法将多目标函数转化为单目标函数, 则八连杆双动拉延压力机内、外滑块执行机构优化设计的目标函数为

$$f(X) = W_1 f_1(X) + W_2 f_2(X) + W_3 f_3(X) \quad (8.6.11)$$

式中

$$f_1(X) = \left\{ \frac{1}{m} \sum_{i=1}^m [\bar{V}(\alpha_i) - \bar{V}_m]^2 \right\}^{\frac{1}{2}}$$

$$f_2(X) = \bar{V}_m$$

$$f_3(X) = \begin{cases} 0 & S_g \geq S_{g1} \\ S_{g1} - S_g & S_g < S_{g1} \end{cases}$$

其中

$$\bar{V}_m = \frac{1}{m} \sum_{i=1}^m \bar{V}(\alpha_i)$$

$$\bar{V}(\alpha_i) = V(\alpha_i) / \omega$$

$$S_g = S(X, \alpha_g)$$

而

$$\alpha_g = \pi + \arctan \frac{A_9}{B_9} - \arccos \frac{R^2 + C_9^2 - l_7^2}{2Rl_7} - \alpha_i$$

这里

$$A_9 = a - l_6 \sin(\psi_6 + \theta_4)$$

$$B_9 = b - l_6 \cos(\psi_6 + \theta_4)$$

$$C_9 = \sqrt{A_9^2 + B_9^2}$$

而

$$\psi_6 = \arctan \frac{A_3}{B_3} + \arccos \frac{l_8^2 + C_3^2 - l_9^2}{2l_8 C_3}$$

式中, a_i 为内滑块压力行程之前, 外滑块压边所需要的提前压紧角; m 为压紧角 α_p ($\alpha_p = \alpha_g - \alpha_1$) 的等分数; a_i 为第 i 个等分点曲柄 R 所在的位置角。

3. 约束条件

由于非独立构件参数的计算, 机构中的曲柄摆杆机构自动成立。考虑到压力机结构的布置、机构成立的条件和传动角的要求等, 可建立如下约束条件:

$$x_{i \min} \leq x_i \leq x_{i \max} \quad (i = 1, 2, \dots, 16)$$

$$l_8 + l_{10} - l_9 \leq c$$

$$b + l_{11} + l_{12} \leq [b + l_{11} + l_{12}]_{\max}$$

$$b + R_1 + l_1 + l_2 \leq [b + R_1 + l_1 + l_2]_{\max}$$

$$R_1 \leq R_{1 \max}$$

$$\begin{aligned}
\overline{OE}_{1\max} &\leq l_9 + l_{10} \\
\overline{OE}_{1\min} &\geq |l_{10} - l_9| \\
\overline{BE}_{\max} &\leq l_3 + l_4 \\
\overline{BE}_{\min} &\geq |l_3 - l_4| \\
\psi_2 + \theta_4 &\leq \pi + \arctan \frac{a}{b} \\
\psi_1 + \theta_4 &\geq \arctan \frac{a}{b} \\
\rho_b + \theta_3 &\leq [\rho_b + \theta_3]_{\max} \\
\arccos \frac{l_6^2 + l_7^2 - (c+R)^2}{2l_6l_7} &\leq \gamma_{\max} \\
\arccos \frac{l_6^2 + l_7^2 - (c-R)^2}{2l_6l_7} &\geq \gamma_{\min} \\
\arccos \frac{l_{10}^2 + l_9^2 - C_{10}^2}{2l_{10}l_9} &\leq \phi_{\max} \\
\arccos \frac{l_{10}^2 + (l_8 + l_9)^2 - a^2 - b^2}{2l_{10}(l_8 + l_9)} &\geq \phi_{\min} \\
\alpha_1 + \alpha_3 &\geq [\alpha_1 + \alpha_3]_{\min}
\end{aligned} \tag{8.6.12}$$

式中

$$\begin{aligned}
C_{10} &= \sqrt{(a - l_8 \sin \psi_2)^2 + (b - l_8 \cos \psi_2)^2} \\
c &= \sqrt{a^2 + b^2} \\
\alpha_3 &= \pi - \arctan \frac{A_9}{B_9} - \arccos \frac{R^2 + C_9^2 - l_7^2}{2RC_9}
\end{aligned}$$

综上所述，八连杆双动拉延压力机内外滑块执行机构优化设计数学模型的标准形式为

$$\min f(X) = \min [W_1 f_1(X) + W_2 f_2(X) + W_3 f_3(X)] \quad (X \in R^{16}) \tag{8.6.13}$$

满足于

$$\begin{aligned}
g_{2i-1}(X) &= x_{i\min} - x_i \leq 0 \quad (i = 1, 2, \dots, 16) \\
g_{2i}(X) &= x_i - x_{i\max} \leq 0 \quad (i = 1, 2, \dots, 16) \\
g_{33}(X) &= l_8 + l_{10} - l_9 - c \leq 0 \\
g_{34}(X) &= b + l_{11} + l_{12} - [b + l_{11} + l_{12}]_{\max} \leq 0
\end{aligned}$$

$$g_{35}(X)=b+R_1+l_1+l_2-[b+R_1+l_1+l_2]_{\max} \leq 0$$

$$g_{36}(X)=R_1-R_{1\max} \leq 0$$

$$g_{37}(X)=\overline{OE}_{1\max}-l_9-l_{10} \leq 0$$

$$g_{38}(X)=|l_{10}-l_9|-\overline{OE}_{1\min} \leq 0$$

$$g_{39}(X)=\overline{BE}_{\max}-l_3-l_4 \leq 0$$

$$g_{40}(X)=|l_3-l_4|-\overline{BE}_{\min} \leq 0$$

$$g_{41}(X)=\psi_2+\theta_4-\pi-\arctan \frac{a}{b} \leq 0$$

$$g_{42}(X)=\arctan \frac{a}{b}-\psi_1-\theta_4 \leq 0$$

$$g_{43}(X)=\rho_b+\theta_3-[\rho_b+\theta_3]_{\max} \leq 0$$

$$g_{44}(X)=\arccos \frac{l_6^2+l_7^2-(c+R)^2}{2l_6l_7}-\gamma_{\max} \leq 0$$

$$g_{45}(X)=\gamma_{\min}-\arccos \frac{l_6^2+l_7^2-(c-R)^2}{2l_6l_7} \leq 0$$

$$g_{46}(X)=\arccos \frac{l_{10}^2+l_9^2-C_{10}^2}{2l_{10}l_9}-\phi_{\max} \leq 0$$

$$g_{47}(X)=\phi_{\min}-\arccos \frac{l_{10}^2+(l_8+l_9)^2-a^2-b^2}{2l_{10}(l_8+l_9)} \leq 0$$

$$g_{48}(X)=[\alpha_1+\alpha_3]_{\min}-\alpha_1-\alpha_3 \leq 0$$

8.6.6 八连杆双动拉延压力机内外滑块执行机构优化设计实例

本节对文献[46]中八连杆板冲压力机 DBS2-1000-3 的内外滑块执行机构进行优化。根据该压力机的具体情况, 取 $750 \leq l_1 \leq 850$, $1700 \leq l_2 \leq 1900$, $700 \leq l_3 \leq 800$, $700 \leq l_5 \leq 800$, $500 \leq l_6 \leq 650$, $650 \leq l_8 \leq 750$, $550 \leq l_{10} \leq 650$, $700 \leq l_{11} \leq 850$, $1700 \leq l_{12} \leq 1950$, $1000 \leq a \leq 1350$, $400 \leq b \leq 700$, $100 \leq e \leq 250$, $45^\circ \leq \theta_1 \leq 70^\circ$, $40^\circ \leq \theta_2 \leq 60^\circ$, $45^\circ \leq \theta_3 \leq 70^\circ$, $50^\circ \leq \theta_4 \leq 70^\circ$, $[b+l_{11}+l_{12}]_{\max}=3350$, $[b+R_1+l_1+l_2]_{\max}=3500$, $R_1 \leq 260$, $[\rho_b+\theta_3]_{\max}=150^\circ$, $\gamma_{\max}=140^\circ$, $\gamma_{\min}=40^\circ$, $\phi_{\max}=140^\circ$, $\phi_{\min}=40^\circ$, $[\alpha_1+\alpha_3]_{\min}=10^\circ$, $\Delta S=0.040$, $S_m=640$, $S_n=800$, $S_g=200$, $\alpha_i=10^\circ$, $W_1=4$, $W_2=1$, $W_3=20$, $m=20$ 。

用复合形法求解, 该压力机执行机构优化设计和原方案构件尺寸及有关数据如表 8.6.1 所示, 其中, 括号中的数据为原方案构件尺寸及有关数据。比较表中的数据可知, 优化设计的执行机构的运动参数 S_n 、 S_g 、 S_m 和 ΔS 较精确地为预先给定的值, 它们的微小误差由数据结尾

误差所引起。并且, 内滑块压力行程的准平均速度 \bar{v}_m 和速度的均匀程度优于原方案, 特别是在压紧角内, 外滑块的波动量仅为 0.03mm, 完全能满足拉延工艺压边允许外滑块的波动量为 0.03~0.05mm 的要求。

表 8.6.1 执行机构优化设计和原方案构件尺寸及有关数据

| | | | |
|----------|--------------------|-------------|---------------------|
| l_1 | 792.531 (800) | e | 191.977 (200) |
| l_2 | 1732.625 (1810) | R | 357.128 (380) |
| l_3 | 727.993 (750) | R_1 | 151.416 (180) |
| L_4 | 882.156 (755) | θ_1 | 46.50814 (58.15) |
| l_5 | 740.142 (730) | θ_2 | 43.71663 (45.88333) |
| l_6 | 532.043 (595) | θ_3 | 45.98792 (52.38333) |
| l_7 | 1249.605 (1265.18) | θ_4 | 69.83939 (64.98333) |
| l_8 | 656.084 (683) | ΔS | 0.030 (0.430) |
| l_9 | 493.289 (502.38) | S_m | 640.000 (638.610) |
| l_{10} | 590.478 (615) | S_n | 800.002 (802.524) |
| l_{11} | 710.161 (773) | S_g | 190.018 (191.449) |
| l_{12} | 1919.134 (1880) | $f_1(X^*)$ | 35.5404 (45.76715) |
| a | 1188.376 (1200) | \bar{v}_m | 131.7816 (138.0369) |
| b | 442.469 (550) | $f(X^*)$ | 273.9432 (321.1057) |

优化的八连杆板冲压力机 $\Pi BS2-1000-3$ 的内、外滑块执行机构的位移和速度曲线如图 8.6.9 所示, 原八连杆板冲压力机 $\Pi BS2-1000-3$ 的内外滑块执行机构的位移和速度曲线如图 8.6.10 所示, 优化的外滑块执行机构和原压力机外滑块执行机构在压紧角内的位移曲线如图 8.6.11 所示。从图 8.6.11 可以明显地看出, 优化的外滑块执行位移的波动量仅为 0.03mm, 完全能满足拉延工艺压边允许外滑块位移的波动量为 0.03~0.05mm 的要求; 而原压力机外滑块位移的波动量约为 0.43mm, 不能满足拉延工艺压边允许外滑块位移的要求。

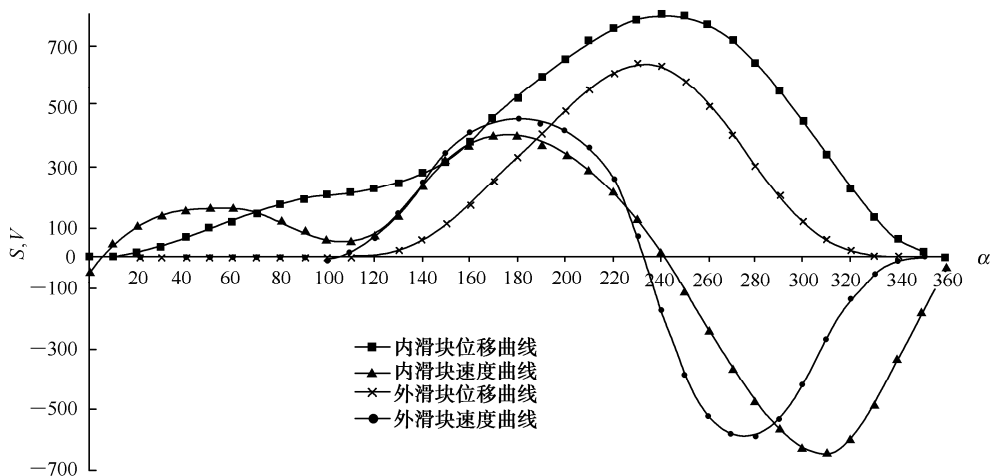


图 8.6.9 优化的执行机构内外滑块位移和速度曲线

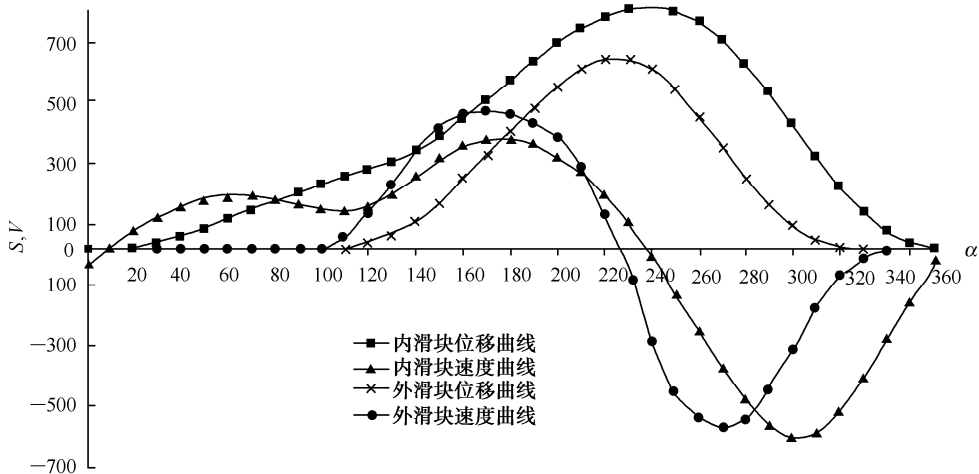


图 8.6.10 原执行机构内外滑块位移和速度曲线

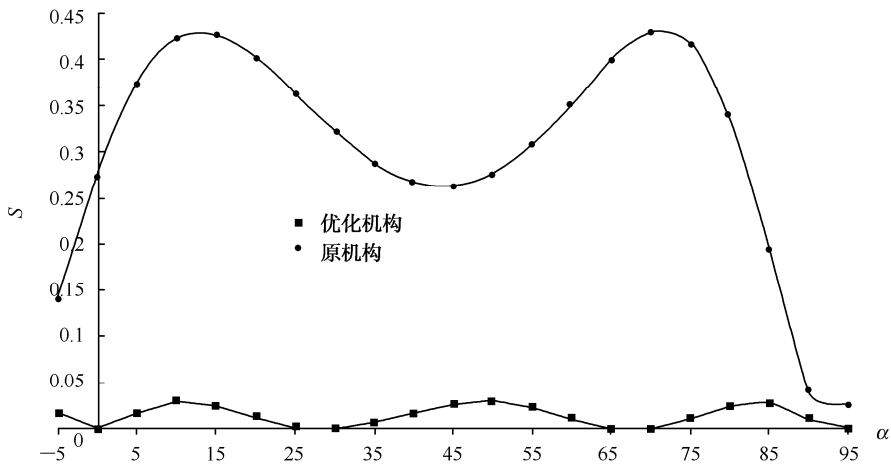


图 8.6.11 原设计与优化外滑块在压紧角内的位移曲线比较

8.7 双动拉延压力机内滑块十杆机构的优化设计^[15,47]

8.7.1 内滑块十杆机构

双动拉延压力机内滑块十杆机构如图 8.7.1 所示。由于该机构具有显著的急回特性，压力行程速度较慢，因此，这种机构非常适合于作拉延压力机内滑块的运动机构。内滑块十杆机构尺寸参数如下：

- O ——曲柄回转中心;
 C ——摆杆 l_3 、 l_4 回转中心;
 M ——连杆与滑块的铰链点;
 R ——曲柄的长度, 即 $R=\overline{OG}$;
 l_1 ——杆 OA 的长度;
 l_2 ——杆 AB 的长度;
 l_3 ——摆杆 BC 的长度;
 l_4 ——摆杆 CD 的长度;
 l_5 ——杆 DE 的长度;
 l_6 ——杆 OE 的长度;
 l_7 ——杆 OF 的长度;
 l_8 ——杆 FH 的长度;
 l_9 ——杆 GH 的长度;
 l_{10} ——连杆 HM 的长度;
 e ——滑块偏置值;
 x_c ——曲柄回转中心与摆杆回转中心的水平尺寸;
 y_c ——曲柄回转中心与摆杆回转中心的垂直尺寸;
 $\theta_1 = \angle BCD$
 $\theta_2 = \angle EOF$

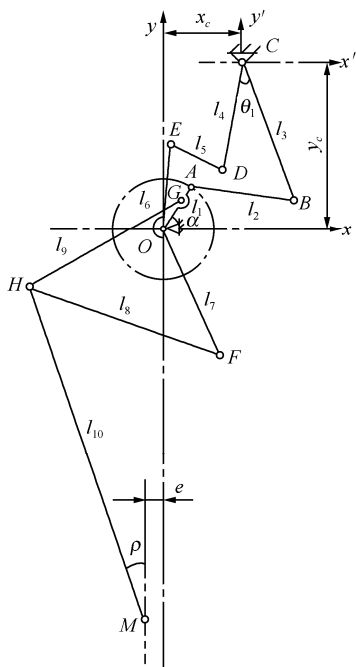


图 8.7.1 内滑块十杆机构简图

8.7.2 非独立机构构件尺寸参数的计算

为了改善拉延压力机内滑块十杆机构的受力条件和防止内滑块在一个周期循环内出现两次向下移动的现象, 取机构参数 R 、 l_9 和 l_{10} 成一条直线时作为内滑块位置的下死点。为了使 R 、 l_9 和 l_{10} 成一条直线, 取构件参数 l_2 为非独立参数。根据拉延压力机内滑块十杆机构的几何关系, l_2 的计算公式为

$$l_2 = \sqrt{(x_c + l_3 \cos \alpha_3 + l_1 \sin \alpha_0)^2 + (y_c + l_3 \sin \alpha_3 + l_1 \cos \alpha_0)^2} \quad (8.7.1)$$

式中

$$\alpha_3 = \alpha_2 + \arccos \frac{C_1^2 + l_4^2 - l_5^2}{2C_1 l_4} + \theta_1$$

$$\alpha_2 = \arctan \frac{l_6 \sin \alpha_1 - y_c}{l_6 \cos \alpha_1 - x_c} \quad (0^\circ \leq \alpha_2 \leq 360^\circ)$$

$$\alpha_1 = \arccos \frac{l_7^2 + (R + l_9)^2 - l_8^2}{2l_7(R + l_9)} + \theta_2 - \alpha_0 - 90^\circ$$

$$\alpha_0 = \arcsin \frac{e}{R + l_9 + l_{10}}$$

$$C_1 = \sqrt{(l_6 \cos \alpha_1 - x_c)^2 + (l_6 \sin \alpha_1 - y_c)^2}$$

α_0 、 α_1 、 α_2 和 α_3 都是当 R 、 l_9 和 l_{10} 成一条直线（即滑块在下死点）时的一些几何角度。 α_0 为构件 R 与铅垂线的夹角； α_1 为 l_6 与 x 轴的夹角； α_2 为 \overline{CE} 与 x' 轴的夹角； α_3 为 l_3 与 x' 轴的夹角。[注： α_2 ($0^\circ \leq \alpha_2 \leq 360^\circ$) 的取值范围与一般数学中正切函数的取值范围不同，要根据分子作为 y 坐标值、分母作为 x 坐标值组成的点 (x, y) 所在的象限来确定，如分子、分母的值分别为 -1 和 -1 ，则 $\alpha_2 = 225^\circ$ ，而不是 $\alpha_2 = 45^\circ$ 。以下角度在 0° 到 360° 范围内取值的反三角函数类同。]

8.7.3 内滑块位移和速度公式的推导

取滑块位移的下死点作为滑块位移的起点，滑块的位移为

$$S = \sqrt{(R + l_9 + l_{10})^2 - e^2} + R \sin \alpha + l_9 \sin \psi - l_{10} \cos \rho \quad (8.7.2)$$

式中

$$\rho = \arcsin \frac{-R \cos \alpha - l_9 \cos \psi - e}{l_{10}}$$

$$\psi = \psi_1 - \psi_2$$

$$\psi_1 = \arctan \frac{l_7 \sin \gamma - R \sin \alpha}{l_7 \cos \gamma - R \cos \alpha} \quad (0^\circ \leq \psi_1 \leq 360^\circ)$$

$$\psi_2 = \arccos \frac{C_2^2 + l_9^2 - l_8^2}{2C_2 l_9}$$

$$C_2 = \sqrt{(l_7 \cos \gamma - R \cos \alpha)^2 + (l_7 \sin \gamma - R \sin \alpha)^2}$$

$$\gamma = \gamma_1 + \gamma_2 - \theta_2$$

$$\gamma_1 = \arctan \frac{y_c + l_4 \sin \beta}{x_c + l_4 \cos \beta} \quad (0^\circ \leq \gamma_1 \leq 360^\circ)$$

$$\gamma_2 = \arccos \frac{C_3^2 + l_6^2 - l_5^2}{2C_3 l_6}$$

$$C_3 = \sqrt{(l_4 \cos \beta + x_c)^2 + (l_4 \sin \beta + y_c)^2}$$

$$\beta = \beta_1 + \beta_2 - \theta_1$$

$$\beta_1 = \arctan \frac{l_1 \sin \alpha - y_c}{l_1 \cos \alpha - x_c} \quad (0^\circ \leq \beta_1 \leq 360^\circ)$$

$$\beta_2 = \arccos \frac{C_4^2 + l_3^2 - l_2^2}{2C_4 l_3}$$

$$C_4 = \sqrt{(l_1 \cos \alpha - x_c)^2 + (l_1 \sin \alpha - y_c)^2}$$

β_1 、 β_2 、 β 、 γ_1 、 γ_2 、 γ 、 ψ_1 、 ψ_2 和 ψ 是当曲柄 R 为 α 时的一些几何角度。其中， $\beta_2 = \angle ACB$ ；

β_1 为 \overline{CA} 与 x' 轴的夹角; β 为 l_4 与 x' 轴的夹角; $\gamma_2=\angle EOD$, γ_1 为 \overline{OD} 与 x 的夹角, γ 为 l_7 与 x 轴的夹角; $\psi_2=\angle FGH$, ψ_1 以 G 点作为坐标原点, \overline{GF} 与 x 轴的平行线的夹角, ψ 以 G 点作为坐标原点; l_9 与 x 轴的平行线的夹角。

滑块的位移 S 对时间 t 求导, 可得滑块的速度为

$$V=\frac{dS}{dt}=R\omega\cos\alpha+l_9\frac{d\psi}{dt}\cos\psi+l_{10}\frac{d\rho}{dt}\sin\rho \quad (8.7.3)$$

式中

$$\frac{d\rho}{dt}=\frac{R\omega\sin\alpha+l_9\frac{d\psi}{dt}\sin\psi}{l_{10}\cos\rho}$$

$$\frac{d\psi}{dt}=\frac{d\psi_1}{dt}-\frac{d\psi_2}{dt}$$

这里

$$\frac{d\psi_1}{dt}=\frac{\cos^2\psi_1\left[l_7^2\frac{d\gamma}{dt}+R^2\omega-Rl_7\left(\frac{d\gamma}{dt}+\omega\right)\cos(\gamma-\alpha)\right]}{(l_7\cos\gamma-R\cos\alpha)^2}$$

$$\frac{d\psi_2}{dt}=\frac{l_7R\left(\frac{d\gamma}{dt}-\omega\right)(l_9\cos\psi_2-C_2)\sin(\gamma-\alpha)}{l_9C_2^2\sin\psi_2}$$

其中

$$\frac{d\gamma}{dt}=\frac{d\gamma_1}{dt}+\frac{d\gamma_2}{dt}$$

而

$$\frac{d\gamma_1}{dt}=\frac{l_4\frac{d\beta}{dt}\cos^2\gamma_1(l_4+x_c\cos\beta+y_c\sin\beta)}{(x_c+l_4\cos\beta)^2}$$

$$\frac{d\gamma_2}{dt}=\frac{l_4\frac{d\beta}{dt}(l_6\cos\gamma_2-C_3)(y_c\cos\beta-x_c\sin\beta)}{l_6C_3^2\sin\gamma_2}$$

式中

$$\frac{d\beta}{dt}=\frac{d\beta_1}{dt}+\frac{d\beta_2}{dt}$$

而

$$\frac{d\beta_1}{dt}=\frac{l_1\omega\cos^2\beta_1(l_1-x_c\cos\alpha-y_c\sin\alpha)}{(l_1\cos\alpha-x_c)^2}$$

$$\frac{d\beta_2}{dt}=\frac{l_1\omega(l_3\cos\beta_2-C_4)(x_c\sin\alpha-y_c\cos\alpha)}{l_3C_4^2\sin\beta_2}$$

这里, ω 为曲柄角速度, 其值为

$$\omega = \frac{d\alpha}{dt}$$

8.7.4 内滑块十杆机构优化设计数学模型的建立

1. 设计变量

在 8.7.2 节中, 机构尺寸参数 l_2 定为非独立参数。双动拉延压力机内滑块十杆机构优化设计的设计变量由机构的独立参数组成, 为了使所有设计方案的内滑块行程和内滑块的压力行程精确地为预先给定的值, 以及便于计算和减少设计变量, 可先将曲柄半径取为单位长度 (即 $R=1$), 而其他构件尺寸取为相对长度 (即构件尺寸与曲柄半径的比值), 求出滑块的相对行程, 然后由滑块行程和滑块的相对行程确定曲柄半径及其他构件尺寸参数。因此, 内滑块十杆机构优化设计的设计变量为

$$X = (x_1, x_2, \dots, x_{14})^T = (l_1, l_3, l_4, l_5, l_6, l_7, l_8, l_9, l_{10}, x_C, y_C, e, \theta_1, \theta_2)^T \quad (8.7.4)$$

2. 目标函数

内滑块在压力行程阶段速度均匀有利于拉延成形工艺, 因此, 要求内滑块在压力行程阶段速度尽量均匀; 还要求减少滑块对导轨的摩擦, 即使滑块对导轨的侧压力尽量小。将双动拉延压力机的拉延压力在整个压力行程中近似地视为常数, 设拉延压力为压机的公称压力 P_g , 同时, 拉延压力机工作时, 连杆 l_{10} 与铅垂线的夹角 ρ 较小, 则有 $\rho \approx \tan \rho$, 把导轨与滑块的摩擦系数 μ 和压机的公称压力 P_g 等常数放到加权因子中, 则滑块对导轨的侧压力的大小等价于 ρ 的大小。分别用函数 $f_1(X)$ 来表示压机在压力行程阶段中内滑块速度的均匀程度、用函数 $f_2(X)$ 来表示在压力行程阶段滑块对导轨的侧压力的大小。双动拉延压力机内滑块十杆机构优化设计是一个多目标函数的优化设计问题, 用线性加权法将多目标函数转化为单目标函数, 则双动拉延压力机内滑块十杆机构优化设计的目标函数为

$$f(X) = W_1 f_1(X) + W_2 f_2(X) \quad (8.7.5)$$

式中

$$f_1(X) = \left\{ \frac{1}{m} \sum_{i=1}^m [\bar{V}(\alpha_i) - \bar{V}_m]^2 \right\}^{\frac{1}{2}}$$

$$f_2(X) = \rho_{\max}$$

其中

$$\bar{V}_m = \frac{1}{m} \sum_{i=1}^m \bar{V}(\alpha_i) = \frac{1}{m} \sum_{i=1}^m \frac{V(\alpha_i)}{\omega}$$

$$\rho_{\max} = \max \{ |\rho(\alpha_i)|, i=1, 2, \dots, m \}$$

式中, m 为在拉延压力机压力行程阶段 S_g 内曲柄 R 所在的位置角的等分点数; α_i 为第 i 个等分点曲柄 R 的位置角; W_1 和 W_2 为加权系数。

3. 约束条件

根据十杆机构成立的条件、传动角的要求和机身结构尺寸的限制,可建立如下约束条件:

$$\begin{aligned}
 l_2 - l_1 &\geq 0 \\
 l_3 - l_1 &\geq 0 \\
 C - l_1 &\geq 0 \\
 l_2 + l_3 - l_1 - C &\geq 0 \\
 C + l_3 - l_1 - l_2 &\geq 0 \\
 C + l_2 - l_1 - l_3 &\geq 0 \\
 \arccos \frac{l_2^2 + l_3^2 - (C - l_1)^2}{2l_2l_3} &\geq \phi_{\min} \\
 \arccos \frac{l_2^2 + l_3^2 - (C + l_1)^2}{2l_2l_3} &\leq \phi_{\max} \\
 l_8 + l_7 - R - l_9 &\geq 0 \\
 |l_9 - R| + l_7 - l_8 &\geq 0 \\
 |l_9 - R| + l_8 - l_7 &\geq 0 \\
 \arccos \frac{l_9^2 + l_8^2 - (l_7 - R)^2}{2l_9l_8} &\geq \chi_{\min} \\
 \arccos \frac{l_9^2 + l_8^2 - (l_7 + R)^2}{2l_9l_8} &\leq \chi_{\max} \\
 \arccos \frac{l_7^2 + l_8^2 - (l_9 - R)^2}{2l_7l_8} &\geq \eta_{\min} \\
 \arccos \frac{l_7^2 + l_8^2 - (l_9 + R)^2}{2l_7l_8} &\leq \eta_{\max} \\
 \frac{|C_1^2 + l_4^2 - l_5^2|}{2C_1l_4} &\leq 1 \\
 l_5 + l_6 - C_{3\max} &\geq 0 \\
 C_{3\min} - |l_5 - l_6| &\geq 0
 \end{aligned} \tag{8.7.6}$$

$$\arccos \frac{l_5^2 + l_6^2 - C_{3\min}^2}{2l_5l_6} \geq \varphi_{\min}$$

$$\arccos \frac{l_5^2 + l_6^2 - C_{3\max}^2}{2l_5l_6} \leq \varphi_{\max}$$

$$\arccos \frac{l_4^2 + l_5^2 - C_{5\min}^2}{2l_4l_5} \geq \vartheta_{\min}$$

$$\arccos \frac{l_4^2 + l_5^2 - C_{5\max}^2}{2l_4l_5} \leq \vartheta_{\max}$$

$$R + l_9 + l_{10} \leq [R + l_9 + l_{10}]_{\max}$$

$$R_{\min} \leq R \leq R_{\max}$$

$$x_{i\min} \leq x_i \leq x_{i\max} \quad (i = 1, 2, \dots, 14)$$

式中, $C_{5\max} = \max C_5$, $C_5 = \sqrt{[l_6 \cos(\gamma_1 + \gamma_2) - x_c]^2 + [l_6 \sin(\gamma_1 + \gamma_2) - y_c]^2}$, $C = \sqrt{x_c^2 + y_c^2}$; ϕ_{\min} 和 ϕ_{\max} 为 $\angle ABC$ 的最小值和最大值 (即 $\angle ABC$ 的取值范围为 $[\phi_{\min}, \phi_{\max}]$); $x_{i\max}$ 为 x_i 的最大值, 其余类推。

综上所述, 双动拉延压力机内滑块十杆机构优化设计数学模型的标准形式为

$$\min f(X) = \min[W_1 f_1(X) + W_2 f_2(X)] \quad (X \in R^{14}) \quad (8.7.7)$$

满足于

$$g_1(X) = l_1 - l_2 \leq 0$$

$$g_2(X) = l_1 - l_3 \leq 0$$

$$g_3(X) = l_1 - C \leq 0$$

$$g_4(X) = l_1 + C - l_2 - l_3 \leq 0$$

$$g_5(X) = l_1 + l_2 - C - l_3 \leq 0$$

$$g_6(X) = l_1 + l_3 - C - l_2 \leq 0$$

$$g_7(X) = \phi_{\min} - \arccos \frac{l_2^2 + l_3^2 - (C - l_1)^2}{2l_2l_3} \leq 0$$

$$g_8(X) = \arccos \frac{l_2^2 + l_3^2 - (C + l_1)^2}{2l_2l_3} - \phi_{\max} \leq 0$$

$$g_9(X) = R + l_9 - l_8 - l_7 \leq 0$$

$$g_{10}(X) = l_8 - |l_9 - R| - l_7 \leq 0$$

$$g_{11}(X) = l_7 - |l_9 - R| - l_8 \leq 0$$

$$g_{12}(X) = \chi_{\min} - \arccos \frac{l_9^2 + l_8^2 - (l_7 - R)^2}{2l_9l_8} \leq 0$$

$$g_{13}(X) = \arccos \frac{l_9^2 + l_8^2 - (l_7 + R)^2}{2l_9l_8} - \chi_{\max} \leq 0$$

$$g_{14}(X) = \eta_{\min} - \arccos \frac{l_7^2 + l_8^2 - (l_9 - R)^2}{2l_7l_8} \leq 0$$

$$g_{15}(X) = \arccos \frac{l_7^2 + l_8^2 - (l_9 + R)^2}{2l_7l_8} - \eta_{\max} \leq 0$$

$$g_{16}(X) = \frac{|C_1^2 + l_4^2 - l_5^2|}{2C_1l_4} - 1 \leq 0$$

$$g_{17}(X) = C_{3\max} - l_5 - l_6 \leq 0$$

$$g_{18}(X) = |l_5 - l_6| - C_{3\min} \leq 0$$

$$g_{19}(X) = \varphi_{\min} - \arccos \frac{l_5^2 + l_6^2 - C_{3\min}^2}{2l_5l_6} \leq 0$$

$$g_{20}(X) = \arccos \frac{l_5^2 + l_6^2 - C_{3\max}^2}{2l_5l_6} - \varphi_{\max} \leq 0$$

$$g_{21}(X) = \vartheta_{\min} - \arccos \frac{l_4^2 + l_5^2 - C_{5\min}^2}{2l_4l_5} \leq 0$$

$$g_{22}(X) = \arccos \frac{l_4^2 + l_5^2 - C_{5\max}^2}{2l_4l_5} - \vartheta_{\max} \leq 0$$

$$g_{23}(X) = R + l_9 + l_{10} - [R + l_9 + l_{10}]_{\max} \leq 0$$

$$g_{24}(X) = R_{\min} - R \leq 0$$

$$g_{25}(X) = R - R_{\max} \leq 0$$

$$g_{25+2i-1}(X) = x_{i\min} - x_i \leq 0 \quad (i = 1, 2, \dots, 14)$$

$$g_{25+2i}(X) = x_i - x_{i\max} \leq 0 \quad (i = 1, 2, \dots, 14)$$

8.7.5 用半惩罚函数法求内滑块十杆机构优化设计的最优解

用第4章4.3节提出的半惩罚函数法求解双动拉延压力机内滑块十杆机构优化设计的最优解, 必须将约束最优化问题中的不等式约束条件分为两类: 第一类约束条件为破坏约束条件会引起计算程序中中断的约束条件, 用 I_1 来表示所有第一类约束条件的下标的集合, 即 $I_1 = \{i: g_i(X) \leq 0, g_i(X) \leq 0 \text{ 为第一类约束条件}\}$; 第二类约束条件为破坏约束条件不会引起计算程序中中断的约束条件, 用 I_2 来表示所有第二类约束条件的下标的集合, 即 $I_2 = \{i: g_i(X) \leq 0, g_i(X) \leq 0 \text{ 为第二类约束条件}\}$ 。

经过分析, 可将双动拉延压力机内滑块十杆机构优化设计中的约束条件分为第一类和第二类约束条件。第一类约束条件的下标的集合 $I_1 = \{1, 2, 3, 4, 5, 6, 9, 10, 11, 16, 17, 18\}$, 第二类约束条件的下标的集合 $I_2 = \{7, 8, 12, 13, 14, 15, 19, 20, 21, \dots, 53\}$ 。

将第二类约束条件按照外点罚函数法来处理, 将其惩罚项加到目标函数中形成罚函数, 而保留第一类约束条件, 形成新的最优化问题, 即

$$\min \varphi(X, \gamma_k) = \min [f(X) + \gamma_k \alpha(X)] \quad (8.7.8)$$

满足于

$$g_i(X) \leq 0 \quad (i \in I_1)$$

其中, γ_k 为外点罚函数法中的惩罚因子, 惩罚项为

$$\alpha(X) = \sum_{i \in I_2} \{\max[0, g_i(X)]\}^2 \quad (8.7.9)$$

式(8.7.8)和式(8.7.9)中的具体函数可由式(8.7.7)求得。

8.7.6 双动拉延压力机内滑块十杆机构的优化设计实例

选用文献[48]中内滑块行程为 850mm、最大拉深行程为 390mm 的十杆机构作为优化设计的数例, 取 $W_1=1$, $W_2=1$, $m=15$ 。

原双动拉延压力机内滑块十杆机构构件尺寸参数为: $R=160\text{mm}$, $l_1=285\text{mm}$, $l_2=510\text{mm}$, $l_3=760\text{mm}$, $l_4=570\text{mm}$, $l_5=320\text{mm}$, $l_6=445\text{mm}$, $l_7=700\text{mm}$, $l_8=950\text{mm}$, $l_9=890\text{mm}$, $l_{10}=1600\text{mm}$, $x_C=380\text{mm}$, $y_C=850\text{mm}$, $e=60\text{mm}$, $\theta_1=32^\circ$, $\theta_2=150^\circ$ 。

原双动拉延压力机内滑块十杆机构的目标函数值为 $f(X_0)=55.883\ 593$, 滑块行程 $S_0=851.512\text{mm}$; 压力机压力行程阶段, 连杆 l_{10} 对滑块的最大压力角 $\rho_{0\max}=9.358\ 261^\circ$ 。

用本节建立的优化设计的数学模型优化后, 双动拉延压力机内滑块十杆机构构件尺寸参数为: $R^*=208.107\text{mm}$, $l_1^*=312.372\text{mm}$, $l_2^*=606.677\text{mm}$, $l_3^*=720.439\text{mm}$, $l_4^*=461.540\text{mm}$, $l_5^*=302.165\text{mm}$, $l_6^*=434.345\text{mm}$, $l_7^*=952.485\text{mm}$, $l_8^*=756.678\text{mm}$, $l_9^*=952.695\text{mm}$, $l_{10}^*=1491.320\text{mm}$, $x_C^*=260.710\text{mm}$, $y_C^*=852.231\text{mm}$, $e^*=23.278\text{mm}$, $\theta_1^*=24.1345^\circ$, $\theta_2^*=159.882^\circ$ 。

优化后双动拉延压力机内滑块十杆机构的目标函数值为 $f(X^*)=23.34176$, 滑块行程 $S_0=850.002\text{mm}$; 压力机压力行程阶段, 连杆 l_{10} 对滑块的最大压力角 $\rho_{\max}^*=5.830957^\circ$ 。

内滑块位移和准速度(曲柄角速度 $\omega=1$ 时, 滑块的速度)如图 8.7.2 所示, 其中曲线 1

和曲线 3 为优化后拉延压力机内滑块位移和准速度曲线，曲线 2 和曲线 4 为原拉延压力机内滑块位移和准速度曲线。

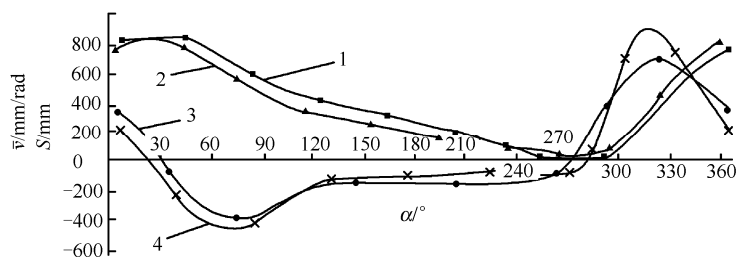


图 8.7.2 内滑块位移和准速度曲线

比较得知，在压力机压力行程 S_g 阶段，优化后双动拉延压力机内滑块速度更加均匀，连杆 l_{10} 对滑块的最大压力角更小。优化的内滑块十杆机构会使压力机拉延工艺更好、滑块对导轨磨损更小。

第9章 最优化方法在拟合公式中的应用

在实际工程中,有许多问题无法用精确的数学公式来表示,科学家们通过实验等手段得到这些问题的有关数据,然后,根据这些实验数据,得到这些问题的近似公式。我们可以借助最优化方法来寻找这些问题的近似公式,或者提高这些问题原有近似公式的计算精度。最优化方法在拟合公式中的应用实例较多,本书只限于讨论作者在拟合公式方面所进行的优化设计实例。这些实例是作者首次独立完成的,本章涉及的优化设计实例有:①有心扇形滑移线场的近似解析式的确定;②假塑性塑料熔体在鱼尾形流道中的流动公式中二项式积分的插值公式的拟合。

9.1 有心扇形滑移线场的近似解析式的拟合^[49]

9.1.1 有心扇形滑移线场的近似解析式

滑移线理论是20世纪20年代初提出来的,滑移线解法是求解理想刚塑性材料平面应变问题的精确理论,但在具体求解时,所建立的滑移线场是近似的,所以,得到的解是近似的。所谓滑移线是指在平面塑性流动区内各点最大切应力方向的轨迹曲线。按滑移线理论可在塑性流动区内画出滑移线场,并容易求出滑移线上的正应力和切应力,从而可求出流动区内各点的应力分布。

滑移线理论假设变形材料是各向同性的刚—塑性体,即在塑性区内忽略了弹性变形,在非塑性区内认为是刚性的。在塑性区内各点的变形抗力是常数,并忽略各点的变形速度和变形温度对变形抗力 σ_s 或最大剪切应力 k 的影响。

滑移线的基本概念可归纳如下^[50]:

- (1) 滑移线是变形体进入塑性状态后,各点的最大切应力方向的轨迹线,是二族正交的曲线族,过一点只能有两条滑移线。
- (2) 滑移线与主应力轨迹线相交成 45° 。
- (3) 滑移线场分布于整个变形体中,一直延伸到变形体的边界。
- (4) 应力场不同,滑移线场也不同。若滑移线场已确定,则相应地应力场也就确定。
- (5) 滑移线场中任一点的最大切应力数值均相同,即 $\tau_{\max}=k$ 。

利用滑移线理论求解塑性成形问题的关键在于建立一个准确的滑移线场。常见的滑移线场有均匀场、简单扇形场、渐开线场、对数螺旋线场、摆线场和有心扇形场等类型。除有心扇形场外,其他各类滑移线场均可用准确的数学解析式来表达,只有有心扇形场无解析式可循,这就给求解这类问题带来极大的不便,给滑移线方法在工程实际中的应

用造成了困难。

为了解决这一问题,已有不少学者在差分法的基础上,提出了一些有心扇形场滑移线场近似解析解。例如,对图 9.1.1 所示的上下对称的有心扇形场,其对称线上各节点的位置 η ($\eta = \frac{y}{b}$) 与扇形场张角 θ 之间的关系就有如下两种近似解析式:

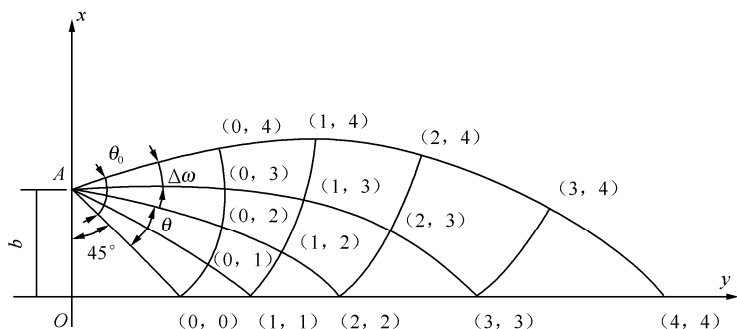


图 9.1.1 上下对称的有心扇形滑移线场

第一种近似解析式为^[51]

$$\theta = \begin{cases} -0.535 + 0.6\eta - 0.065\eta^2 & \theta=0 \sim \frac{\pi}{4} \\ 0.15 + 0.2\eta - 0.007\eta^2 & \theta=\frac{\pi}{4} \sim \frac{\pi}{2} \end{cases} \quad (9.1.1)$$

或写成

$$\eta = \begin{cases} 4.615 - \sqrt{13.071 - 15.385\theta} & \theta=0 \sim \frac{\pi}{4} \\ 14.286 - \sqrt{225.51 - 142.86\theta} & \theta=\frac{\pi}{4} \sim \frac{\pi}{2} \end{cases} \quad (9.1.2)$$

第二种近似解析式为^[52]

$$\theta = 0.6 \ln \eta \quad (9.1.3)$$

或写成

$$\eta = e^{1.67\theta} \quad (9.1.4)$$

为了了解这些近似解析式的准确程度,作者用差分法对上述有心扇形滑移线场进行了计算,并将计算结果与上述两种近似解析式进行比较(见表 9.1.1)。发现这两种近似解析式精度都不高,另外,第一种近似解析式的适用范围窄。

表 9.1.1 有心扇形滑移线差分解和近似解析式计算精度比较

| $\theta(^{\circ})$ | $\Delta\omega=0.05^{\circ}$ 按差分计算 | 按式 (9.1.11) 计算 | | 按式 (9.1.2) 计算 | | 按式 (9.1.4) 计算 | |
|--------------------|--------------------------------------|----------------|-------|----------------|-------|----------------|-------|
| | | $\eta(\theta)$ | 误差 | $\eta(\theta)$ | 误差 | $\eta(\theta)$ | 误差 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 5 | 1.182606 | 1.182761 | 0.01 | 1.190703 | 0.68 | 1.156890 | -2.17 |
| 10 | 1.383321 | 1.383824 | 0.04 | 1.392671 | 0.68 | 1.338393 | -3.25 |
| 15 | 1.605407 | 1.606291 | 0.06 | 1.608173 | 0.17 | 1.548373 | -3.55 |
| 20 | 1.852620 | 1.853785 | 0.06 | 1.840360 | -0.66 | 1.791297 | -3.31 |
| 25 | 2.129293 | 2.130546 | 0.06 | 2.093837 | -1.67 | 2.072333 | -2.68 |
| 30 | 2.440447 | 2.441530 | 0.04 | 2.375822 | -2.65 | 2.397460 | -1.76 |
| 35 | 2.791904 | 2.792536 | 0.02 | 2.698856 | -3.33 | 2.773597 | -0.66 |
| 40 | 3.190435 | 3.190340 | -0.00 | 3.088781 | -3.19 | 3.208745 | 0.57 |
| 45 | 3.643925 | 3.642870 | -0.03 | 3.621423 | -0.62 | 3.712163 | 1.87 |
| 50 | 4.161563 | 4.159394 | -0.05 | 4.243612 | 1.97 | 4.294563 | 3.20 |
| 55 | 4.754081 | 4.750753 | -0.07 | 4.884800 | 2.75 | 4.968336 | 4.51 |
| 60 | 5.434008 | 5.429620 | -0.08 | 5.573049 | 2.56 | 5.747816 | 5.77 |
| 65 | 6.216003 | 6.210817 | -0.08 | 6.320546 | 1.68 | 6.649588 | 6.98 |
| 70 | 7.117219 | 7.111676 | -0.08 | 7.145878 | 0.40 | 7.692839 | 8.09 |
| 75 | 8.157731 | 8.152466 | -0.06 | 8.080021 | -0.95 | 8.899764 | 9.10 |
| 80 | 9.361063 | 9.356885 | -0.04 | 9.182383 | -1.91 | 10.29604 | 9.99 |
| 85 | 10.75479 | 10.75265 | -0.02 | 10.60097 | -1.43 | 11.91139 | 10.75 |
| 90 | 12.37126 | 12.37217 | 0.01 | 13.23180 | 6.96 | 13.78016 | 11.39 |
| 95 | 14.24842 | 14.25334 | 0.03 | | | 15.94212 | 11.89 |
| 100 | 16.43078 | 16.44049 | 0.06 | | | 18.44327 | 12.25 |
| 105 | 18.97065 | 18.98542 | 0.08 | | | 21.33683 | 12.47 |
| 110 | 21.92944 | 21.94875 | 0.09 | | | 24.68436 | 12.56 |
| 115 | 25.37932 | 25.40134 | 0.09 | | | 28.55708 | 12.52 |
| 120 | 29.40512 | 29.42607 | 0.07 | | | 33.03739 | 12.35 |
| 125 | 34.10660 | 34.11981 | 0.04 | | | 38.22060 | 12.06 |
| 130 | 39.60101 | 39.59591 | -0.01 | | | 44.21702 | 11.66 |
| 135 | 46.02637 | 45.98686 | -0.09 | | | 51.15421 | 11.14 |
| 最大相对误差 | | 0.09% | | 6.96% | | 12.56% | |
| 误差波动范围 | | -0.09~0.09% | | -3.33~6.96% | | -3.55~12.56% | |

下面，讨论如何在差分法的基础上，利用优化方法寻找一种高精度有心扇形滑移线近似解析解。

9.1.2 有心扇形场的差分公式

在图 9.1.1 中, 令, $AO=b=1$ 。

圆弧上各点 $(0,n)$ 的坐标为

$$\begin{cases} y_{0,n}=\sqrt{2}\sin(45^\circ+n\Delta\omega) \\ x_{0,n}=1-\sqrt{2}\cos(45^\circ+n\Delta\omega) \end{cases} \quad (9.1.5)$$

令

$$\omega_{0,0}=-45^\circ \quad (9.1.6)$$

则

$$\begin{cases} \omega_{m,n}=(n-m)\Delta\omega+\omega_{0,0} \\ \omega_{m,n-1}=(n-m-1)\Delta\omega+\omega_{0,0} \\ \omega_{m-1,n}=(n-m+1)\Delta\omega+\omega_{0,0} \end{cases} \quad (9.1.7)$$

场中任意一点 (m,n) 的坐标 $(y_{m,n},x_{m,n})$ 按下述差分公式计算:

$$\begin{cases} y_{m,n}=\frac{x_{m,n-1}-x_{m-1,n}+y_{m-1,n}\tan\frac{\omega_{m,n}+\omega_{m-1,n}}{2}+y_{m,n-1}\cot\frac{\omega_{m,n}+\omega_{m,n-1}}{2}}{\tan\frac{\omega_{m,n}+\omega_{m-1,n}}{2}+\cot\frac{\omega_{m,n}+\omega_{m,n-1}}{2}} \\ x_{m,n}=x_{m-1,n}+(y_{m,n}-y_{m-1,n})\tan\frac{\omega_{m,n}+\omega_{m-1,n}}{2} \end{cases} \quad (9.1.8)$$

作者对有心扇形滑移线采用上述公式, 分别按 $\Delta\omega=0.1^\circ$ 和 $\Delta\omega=0.05^\circ$ 进行了计算。发现它们之间的绝对误差仅为 0.00001, 这说明按 $\Delta\omega=0.05^\circ$ 进行差分计算已具有了相当的精度。水平对称线上节点 η ($\eta=\frac{y}{b}$) 的差分结果列于表 9.1.1 中。

9.1.3 用优化方法确定有心扇形滑移近似解析式

在上述差分数值计算的基础上, 作者根据问题的特点, 选用如下函数作为有心扇形滑移线水平对称线上节点坐标与张角函数关系的近似解析式:

$$\eta=ae^{b\theta}+c\theta+1-a \quad (9.1.9)$$

式中, a 、 b 和 c 为待定常数, 这些待定常数用优化方法来确定。

用最大相对误差作为优化的目标函数, 即

$$\min f(a, b, c) = \min \left\{ \max \left[\frac{\eta(\theta_i) - \bar{\eta}(\theta_i)}{\bar{\eta}(\theta_i)}, 1 \leq i \leq n \right] \right\} \quad (9.1.10)$$

式中, n ——有心扇形张角 θ_0 的等分数;

$\eta(\theta_i)$ ——由式(9.1.9)计算的 η 值;

$\bar{\eta}(\theta_i)$ ——由差分公式(9.1.8)计算的 η 值。

通过优化方法确定式(9.1.10)中系数 a 、 b 和 c , 得到有心扇形滑移线近似解析式为

$$\eta = 0.638e^{1.793\theta} + 0.856\theta + 0.362 \quad (9.1.11)$$

为了验证用优化方法确定的有心扇形滑移线近似解析式的精度, 将按式(9.1.11)计算的结果列于表 9.1.1 中。由表 9.1.1 可知, 这种近似解析式的精度高, 误差稳定, 在 $\theta = 0 \sim 135^\circ$ 范围内, 最大相对误差在 0.09% 以下。而第一种近似解析式仅在 $\theta = 0 \sim 90^\circ$ 范围内, 采用两个公式, 其最大相对误差达到 6.96%。第二种近似解析解的最大相对误差高达 12.56%。

从上面讨论中可以看出, 用优化方法来确定有心扇形滑移线近似解析式的精度非常高, 其最大相对误差不到千分之一。

9.1.4 有心扇形滑移线场的近似解析式的应用

以平冲头压入有限厚度板坯的滑移线场为例, 见图 9.1.2 所示。用上节有心扇形滑移线近似解析式求压入力 $\frac{p}{2k}$ 的解析解。

由图 9.1.2 中 e 和 g 点的应力状态可知

$$\sigma_{mg} = \sigma_{xg} - k \quad (9.1.12)$$

$$\sigma_{me} = -p + k \quad (9.1.13)$$

式中, σ_{mg} 和 σ_{me} 分别为 g 点和 e 点的平均应力。

由汉基应力方程可知

$$\sigma_{mg} - 2k\theta = \sigma_{me} + 2k\theta \quad (9.1.14)$$

由式(9.1.12)、式(9.1.13)和式(9.1.14)得

$$\sigma_{xg} = -p + 2k(1 + 2\theta) \quad (9.1.15)$$

式中, σ_{xg} 为沿对称线 $1 \leq \eta \leq \eta_0$ 区段各点的水平应力。

在 $0 \leq \eta \leq \eta_0$ 区段为均匀场, 其应力为

$$\sigma_x = -p + k \quad (9.1.16)$$

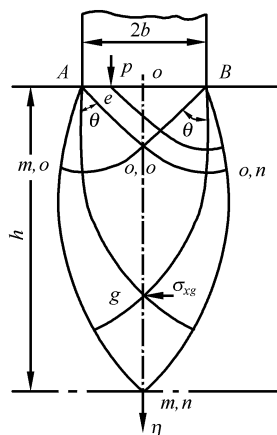


图 9.1.2 有心扇形滑移线场应用实例

因为板坯并未受水平方向的外力, 故有

$$\int_0^{\eta_0} \sigma_x d\eta = 0 \quad (9.1.17)$$

将式 (9.1.15) 和式 (9.1.16) 代入上式, 分段积分, 整理后可得

$$\frac{p}{2k} = 1 + \frac{2}{\eta_0} \int_1^{\eta_0} \theta d\eta \quad (9.1.18)$$

由式 (9.1.11) 可得

$$d\eta = (0.638 \times 1.793 e^{1.793\theta} + 0.856) d\theta \quad (9.1.19)$$

将式 (9.1.19) 代入式 (9.1.18), 可得

$$\begin{aligned} \frac{p}{2k} &= 1 + \frac{2}{\eta_0} \int_0^{\theta_0} \theta (0.638 \times 1.793 e^{1.793\theta} + 0.856) d\theta \\ &= \frac{2}{\eta_0} \left[\frac{0.638}{1.793} (1.793\theta_0 e^{1.793\theta_0} - e^{1.793\theta_0} + 1) + 0.828\theta_0^2 \right] + 1 \end{aligned} \quad (9.1.20)$$

在本例中, $\theta_0 = \frac{5\pi}{12}$, 将 $\theta_0 = \frac{5\pi}{12}$ 代入式 (9.1.11), 可得 $\eta_0 = 8.152465736$, 将 θ_0 和 η_0 的值

代入式 (9.1.20) 求得 $\frac{p}{2k} = 2.496517827$; 按 $\Delta\omega = 0.05^\circ$ 用差分方法求得 η , 再按式 (9.1.18)

进行数值积分求得 $\frac{p}{2k} = 2.49792$, 两者比较, 其相对误差仅有 0.056%, 这说明用近似解析式

求得的压力 $\frac{p}{2k}$ 的解析解的计算精度非常高。

9.2 假塑性塑料熔体在鱼尾形流道中的近似流动公式的拟合^[53]

9.2.1 假塑性塑料熔体在鱼尾形流道中的流动公式的推导

在注射成型中, 大多数聚合物熔体都是非牛顿流体, 且它们的大多数又都近似服从 Ostwald-De Waele 提出的指数流动规律^[54], 即

$$\tau = K \dot{\gamma}^n \quad (9.2.1)$$

式中, K ——与聚合物和温度有关的常数, 它反映聚合物熔体的黏稠性, 称为稠度系数;

n ——与聚合物和温度有关的常数, 它反映聚合物熔体偏离牛顿流体的程度, 称为非牛顿指数;

$\dot{\gamma}$ ——切应变速率;

τ ——剪切应力。

式 (9.2.1) 可改写为

$$\tau = \eta_a \dot{\gamma} \quad (9.2.2)$$

其中

$$\eta_a = K \dot{\gamma}^{n-1} \quad (9.2.3)$$

这里, η_a 称为聚合物的表观黏度。

在聚物流变学理论中, 凡是服从指数流动规律的非牛顿流体, 统称为黏性液体。根据非牛顿指数 n 的取值, 聚合物熔体可分为膨胀性流体和假塑性流体等。当 $n < 1$ 时, 称为假塑性流体。在注射成型中, 绝大多数热塑性塑料属于假塑性流体。

所谓鱼尾形流道, 是指既具有厚度方向锥度 2θ , 又具有宽度方向锥度 2ϕ 的双线性变化的矩形狭缝截面通道 (如图 9.2.1 所示), 它是注塑模设计中常用的一种浇道。在图 9.2.1 中, W_1 、 H_1 为流道入口处的宽度和高度, W_2 、 H_2 为流道出口处的宽度和高度, L 为流道的长度。

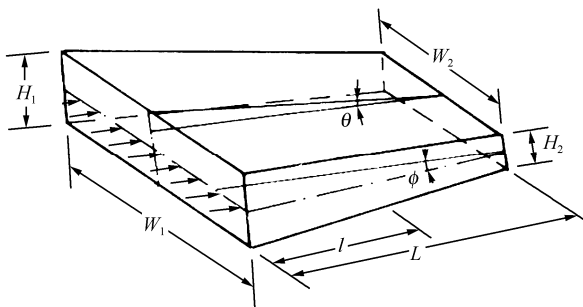


图 9.2.1 鱼尾形流道

由式 (9.2.1) 可以推出塑料熔体在矩形狭缝截面流道中压力降 ΔP 和流量 Q 的关系为^[55]

$$\Delta P = \left(\frac{4n+2}{n} \right)^n Q^n \frac{2KL}{W^n h^{2n+1}} \quad (9.2.4)$$

式中, L 、 W 和 H 分别为矩形狭缝截面流道的长度、宽度和高度。

由式 (9.2.4) 可得鱼尾形流道在 l 处 dl 长度上的压力降 dP 和流量 Q 的关系为

$$dP = 2K \left(\frac{4n+2}{n} \right)^n Q^n \frac{dl}{W^n h^{2n+1}} \quad (9.2.5)$$

式中, W 和 h 分别为鱼尾形流道 l 处横截面的宽度和高度, 它们与 l 的关系为

$$h = H_1 - 2l \tan \theta \quad (9.2.6)$$

$$W = W_1 - 2l \tan \phi \quad (9.2.7)$$

由式 (9.2.6) 和式 (9.2.7) 得

$$h=a+bW \quad (9.2.8)$$

式中

$$b=\frac{\tan \theta}{\tan \phi}=\frac{H_1-H_2}{W_1-W_2} \quad (9.2.9)$$

$$a=H_1-bW_1 \quad (9.2.10)$$

由式 (9.2.7) 可得

$$dl=-\frac{\cot \phi}{2}dW=\frac{L}{W_2-W_1}dW \quad (9.2.11)$$

将式 (9.2.8) 和式 (9.2.11) 代入式 (9.2.5), 得

$$dP=\frac{2KL}{W_2-W_1}\left(\frac{4n+2}{n}\right)^n Q^n \frac{dW}{W^n(a+bW)^{2n+1}} \quad (9.2.12)$$

上式两边积分, 得

$$\Delta P=\frac{2KL}{W_2-W_1}\left(\frac{4n+2}{n}\right)^n Q^n \int_{W_1}^{W_2} \frac{dW}{W^n(a+bW)^{2n+1}} \quad (9.2.13)$$

为了求得假塑性流体在鱼尾形流道中塑料熔体的压力降 ΔP 和流量 Q 的关系式, 还需求定积分 $\int_{W_1}^{W_2} \frac{dW}{W^n(a+bW)^{2n+1}}$ 的值, 由微积分中牛顿-莱布尼兹公式可知, 求得不定积分的原函数后, 很容易求得定积分的值。因此, 为了节省篇幅, 下面, 仅讨论不定积分 $\int \frac{dW}{W^n(a+bW)^{2n+1}}$ 的求解。

分如下四种情况讨论不定积分 $\int \frac{dW}{W^n(a+bW)^{2n+1}}$ 的求解。

(1) 当 $a=0$ (即 $H_1/W_1=H_2/W_2$) 时

$$\int \frac{dW}{W^n(a+bW)^{2n+1}} = -\frac{1}{3nb^{2n+1}W^{3n}}+c \quad (9.2.14)$$

式中, c 为积分常数, 下同。

(2) 当 $b=0$ (即宽楔形流道) 时

$$\int \frac{dW}{W^n(a+bW)^{2n+1}} = \begin{cases} -\frac{1}{(n-1)a^{2n+1}W^{n-1}}+c & n \neq 1 \\ \frac{\ln W}{a^3}+c & n=1 \end{cases} \quad (9.2.15)$$

(3) 当 $n=0$ 时

$$\int \frac{dW}{W^n(a+bW)^{2n+1}} = \frac{\ln(a+bW)}{b} + c \quad (9.2.16)$$

(4) 当 $a \neq 0$ 、 $b > 0$ 和 $n \neq 0$ 时, $\int \frac{dW}{W^n(a+bW)^{2n+1}}$ 属于二项式积分, 由契比协夫定理知, 仅下列三种情形可化为有理函数的积分。

第一种情形, $2n+1$ 为整数, 即 $n \in \{1/2, 1, 3/2, \dots\}$;

第二种情形, $n+1$ 为整数, 即 $n \in \{1, 2, 3, \dots\}$;

第三种情形, $3n+2$ 为整数, 即 $n \in \{1/3, 2/3, 1, \dots\}$ 。

由于绝大多数塑料熔体的非牛顿指数 $n < 1$, 而 $n=1$ 为牛顿流体的特殊情形, 因此, 下面我们仅讨论 $n \in \{1/3, 1/2, 2/3, 1\}$ 时, $\int \frac{dW}{W^n(a+bW)^{2n+1}}$ 的求解。

① 当 $n=1/3$ 时

$$\int \frac{dW}{W^n(a+bW)^{2n+1}} = \frac{3}{2a} \left(\frac{W}{a+bW} \right)^{2/3} + c \quad (9.2.17)$$

② 当 $n=1/2$ 时

$$\int \frac{dW}{W^n(a+bW)^{2n+1}} = \begin{cases} \frac{\sqrt{W}}{a(a+bW)} + \frac{1}{a\sqrt{ab}} \arctan \sqrt{\frac{bW}{a}} + c & a > 0 \\ \frac{\sqrt{W}}{a(a+bW)} + \frac{1}{2a\sqrt{-ab}} \ln \frac{\sqrt{bW} - \sqrt{-a}}{\sqrt{bW} + \sqrt{-a}} + c & a < 0 \end{cases} \quad (9.2.18)$$

③ 当 $n=2/3$ 时

$$\int \frac{dW}{W^n(a+bW)^{2n+1}} = -\frac{3b}{4a^2 \left(\frac{a+bW}{W} \right)^{4/3}} + \frac{3}{a^2 \left(\frac{a+bW}{W} \right)^{1/3}} + c \quad (9.2.19)$$

④ 当 $n=1$ 时

$$\int \frac{dW}{W^n(a+bW)^{2n+1}} = \frac{1}{a^3} \ln \frac{W}{a+bW} + \frac{1}{a^2(a+bW)} + \frac{1}{2a(a+bW)^2} + c \quad (9.2.20)$$

9.2.2 用优化方法确定插值公式

从上节讨论知, 只有当塑料熔体的非牛顿指数 $n \in \{0, 1/3, 1/2, 2/3, 1, \dots\}$ 时, 才能用公式求

出定积分 $\int_{W_1}^{W_2} \frac{dW}{W^n(a+bW)^{2n+1}}$ 的值。对于 n 为一般值时, 当然, 可以用数值积分的方法求解

$\int_{W_1}^{W_2} \frac{dW}{W^n(a+bW)^{2n+1}}$ 。下面, 我们不讨论用数值积分的方法求解 $\int_{W_1}^{W_2} \frac{dW}{W^n(a+bW)^{2n+1}}$, 而讨论用优化方法确定 $\int_{W_1}^{W_2} \frac{dW}{W^n(a+bW)^{2n+1}}$ 积分的插值公式中的待定常数, 从而得到 $\int_{W_1}^{W_2} \frac{dW}{W^n(a+bW)^{2n+1}}$ 积分的插值公式。

对于 $n \notin \{0.1/3, 1/2, 2/3, 1, \dots\}$, 通过 $n \in \{0.1/3, 1/2, 2/3, 1, \dots\}$ 中两个有精确积分公式的值进行插值。例如, 对于 $n=0.4$ 的定积分 $\int_{W_1}^{W_2} \frac{dW}{W^n(a+bW)^{2n+1}}$ 的计算, 用 $n=1/3$ 和 $n=1/2$ 两个有精确积分公式 (9.2.17) 和公式 (9.2.18) 的值进行插值。根据被积函数的特点, 选用如下形式的插值公式:

$$f(n) = \int_{W_1}^{W_2} \frac{dW}{W^n(a+bW)^{2n+1}} \approx \frac{A}{\overline{W}^{x_1 n + x_2}} + \frac{B}{(a+bW)^{x_3 n + x_4}} \quad (9.2.21)$$

式中, $\overline{W} = \frac{W_1 + W_2}{2}$; x_1, x_2, x_3 和 x_4 作为变量用优化方法确定; A 和 B 为待定常数, 通过相邻两个精确解与插值公式的值相等来确定。

为了确定插值公式 (9.2.21) 中系数 x_1, x_2, x_3 和 x_4 , 选用如下 7 个数例。

例 1. $W_1=100, H_1=10, W_2=60, H_2=5$ 。

例 2. $W_1=150, H_1=10, W_2=60, H_2=5$ 。

例 3. $W_1=250, H_1=15, W_2=60, H_2=5$ 。

例 4. $W_1=250, H_1=25, W_2=60, H_2=5$ 。

例 5. $W_1=100, H_1=10, W_2=90, H_2=5$ 。

例 6. $W_1=50, H_1=5, W_2=30, H_2=2$ 。

例 7. $W_1=40, H_1=3, W_2=10, H_2=1$ 。

用辛卜生数值积分方法求得的定积分 $\int_{W_1}^{W_2} \frac{dW}{W^n(a+bW)^{2n+1}}$ 的值如表 9.2.1 所示。选用插值函数与定积分数值积分的值的相对误差函数作为参数优化设计的目标函数, 其具体的目标函数为

$$\min f(X) = \min \sum_{j=1}^7 \sum_{i=1}^7 \left| \frac{f_i(X, n_j) - \bar{f}_i(n_j)}{\bar{f}_i(n_j)} \right| \quad (9.2.22)$$

式中, $\bar{f}_i(n_j)$ 为当塑料熔体的非牛顿指数为 n_j , 第 i 例按数值积分计算的定积分 $\int_{W_1}^{W_2} \frac{dW}{W^n(a+bW)^{2n+1}}$ 的值; $f_i(X, n_j)$ 为用插值公式 (9.2.21) 计算的 $\int_{W_1}^{W_2} \frac{dW}{W^n(a+bW)^{2n+1}}$ 的值; $n_j \in \{n_1, n_2, n_3, n_4, n_5, n_6, n_7\} = \{0.2, 0.3, 0.4, 0.6, 0.7, 0.8, 0.9\}$ 。

表 9.2.1 鱼尾形流道中 $f(n)$ 的计算值

| n | 例 1 | 例 2 | 例 3 | 例 4 | 例 5 | 例 6 | 例 7 |
|----------|--------------------|--------------------|---------------------|--------------------|--------------------|--------------------|--------------------|
| 0 | 5.5452 | 12.4767 | 20.8736 | 15.2897 | 1.3863 | 6.1086 | 16.4792 |
| 0.2 | 1.0705 (1.0772) | 2.3080 (2.3320) | 3.4 127 (3.4522) | 2.3160 (2.3117) | 0.2563 (0.2584) | 1.8784 (1.8890) | 7.3743 (7.3393) |
| 0.3 | 0.4724 (0.4735) | 0.9992 (1.0031) | 1.4023 (1.4083) | 0.9252 (0.9245) | 0.1105 (0.1108) | 1.0487 (1.0509) | 5.0113 (5.0023) |
| 1/3 | 0.3599 | 0.7566 | 1.0450 | 0.6839 | 0.0835 | 0.8644 | 4.4158 |
| 0.4 | 0.2091 (0.2093) | 0.4345 (0.4354) | 0.5822 (0.5836) | 0.3756 (0.3754) | 0.0477 (0.0478) | 0.5881 (0.5887) | 3.4402 (3.4360) |
| 1/2 | 0.0928 | 0.1897 | 0.2442 | 0.1548 | 0.0207 | 0.3312 | 2.3849 |
| 0.6 | 0.0413 (0.0413) | 0.0832 (0.0833) | 0.1034 (0.1036) | 0.0647 (0.0647) | 0.0090 (0.0089) | 0.1873 (0.1875) | 1.6688 (1.6668) |
| 2/3 | 0.0241 | 0.0481 | 0.0586 | 0.0364 | 0.0051 | 0.1284 | 1.3217 |
| 0.7 | 0.0158 (0.0158) | 0.0366 (0.0367) | 0.0442 (0.0443) | 0.0274 (0.0274) | 0.0039 (0.0039) | 0.1064 (0.1065) | 1.1779 (1.1761) |
| 0.8 | 0.0083 (0.0083) | 0.0162 (0.0163) | 0.0190 (0.0191) | 0.0117 (0.0117) | 0.0017 (0.0017) | 0.0607 (0.0609) | 0.8383 (0.8351) |
| 0.9 | 0.0037 (0.0038) | 0.0072 (0.0072) | 0.0083 (0.0083) | 0.0051 (0.0051) | 0.0007 (0.0007) | 0.0347 (0.0349) | 0.6011 (0.5993) |
| 1 | 0.0017 | 0.0032 | 0.0036 | 0.0022 | 0.0003 | 0.0200 | 0.4341 |
| δ | 2.7% | 1.0% | 1.2% | 0.2% | 0.8% | 0.6% | 0.5% |

注：① 所有 $f(n)$ 的值均为负数，为了简洁，表中数据统一省略了负号；

② 当 $n \in \{0, 1/3, 1/2, 2/3, 1, \dots\}$ 时，用数值积分和推导的公式 (9.2.17) ~ 式 (9.2.20) 计算的定积分的值完全相同，为了简洁，表中只填写了一个数据；

③ 括号中数据为用插值公式 (9.2.23) 的计算值。

用以上 7 个数例的数据和优化方法求解式 (9.2.22) 来确定插值公式中 x_1 、 x_2 、 x_3 和 x_4 ，并对

变量适当取整后，得到定积分 $\int_{w_1}^{w_2} \frac{dW}{W^n(a+bW)^{2n+1}}$ 的插值公式为

$$f(n) = \frac{A}{W^{1.6n+3}} + \frac{B}{(a+bW)^{4n+5}} \quad (9.2.23)$$

用以上插值公式 (9.2.23) 计算的 7 个数例定积分 $\int_{w_1}^{w_2} \frac{dW}{W^n(a+bW)^{2n+1}}$ 的值亦列于表 9.2.1 中。比较表中的数据可知，通过优化方法确定的定积分 $\int_{w_1}^{w_2} \frac{dW}{W^n(a+bW)^{2n+1}}$ 近似插值公式的计算精度是非常高的，最大相对误差 $\delta=2.7\%$ 。

对于 $n \notin \{0, 1/3, 1/2, 2/3, 1\}$, 由式 (9.2.13) 和式 (9.2.23), 可得假塑性流体在鱼尾形流道中塑料熔体的压力降 ΔP 和流量 Q 的近似关系式为

$$\Delta P = \frac{2KL}{W_2 - W_1} \left(\frac{4n+2}{n} \right)^n Q^n \left[\frac{A}{\bar{W}^{1.6n+3}} + \frac{B}{(a+b\bar{W})^{4n+5}} \right] \quad (9.2.24)$$

对于 $n \in \{0, 1/3, 1/2, 2/3, 1\}$, 假塑性流体在鱼尾形流道中塑料熔体的压力降 ΔP 和流量 Q 有精确的关系式, 这些精确的关系式可由式 (9.2.13) ~ 式 (9.2.20) 求得, 这些精确的关系式就不一一列出。

附录 A 多层压配组合挤压凹模疲劳 强度优化设计 C++ 程序

1. 源程序

```
//This is a program of optimum design of multi-layer composite cold extrusion die
#include <iostream.h>
#include <iomanip.h>
#include <math.h>
#include <stdlib.h>
double sss[3]={2000,2000,2000};
double ssc[3]={2400,2400,2400};
double ssb[3]={2600,2600,2600};
double ssf0[3]={1200,1200,1200};
double suf[3]={0.92,0.92,0.92};
double rmin[3]={10,10,10};
double rmax[3]={880,880,880};
double e1=210000,e2=210000,e3=210000;
double v1=0.3,v2=0.3,v3=0.3;
double ds=0.8;
double r1=70,r4=650,r3=280,r2=500,p1=1200,csafe=1.2,d0=9;
// one-layer data
// const int nn=1,mm=4;
// double x[nn]={350};
// double bo[nn][2]={ {50,700} };
// two-layer data
// const int nn=3,mm=8;
// double x[nn]={97,560,229};
// double bo[nn][2]={ {68,140},{80,200},{200,400} };
// three-layer data
const nn=5,mm=12;
double x[nn]={80,120,700,32,195};
double bo[nn][2]={ {68,140},{90,250},{90,800},{50,600},{50,400} };
const int n2=100*nn;
int result=0,nlayer;
const double ee=0.000005,eb=0.001;
```

```

char yy;
const char p1r4='r';
double ff,xx,df;
double xk[nn],ga[mm],p[nn][nn];
int subject() {
    int kk=0;
    int ii;
    double a1,c1,a2,c2,ap1,cp1,ap2,cp2,ap3,cp3,p2,p3,dd,ssf[3];
    double aa1,bb1,cc1,dd1,aa2,bb2,cc2,dd2,aa3,bb3,cc3,dd3,cc;
    double stressr[3][2],stresst[3][2],strssa,strssar,strssat,strssm;
// one-layer
    if (nlayer==1) {
        if (yy==p1r4) ff=x[0];
        else ff=-x[0];
        if (yy==p1r4) r2=x[0];
        else p1=x[0];
        ga[0]=-r2+r1+rmin[0];
        ga[1]=r2-r1-rmax[0];
        stressr[0][0]=0;
        stressr[0][1]=-p1;
        stresst[0][0]=0;
        stresst[0][1]=p1*(r2*r2+r1*r1)/(r2*r2-r1*r1);
        if (result) {
            if (yy==p1r4) cout<<setw(12)<<"r2"<<setw(12)<<"p1";
            else cout<<setw(12)<<"p1"<<setw(12)<<"r2";
            cout<<endl<<endl;
            for (ii=0;ii<nn;ii++) {
                cout<<setw(12)<<x[ii];
            }
            if (yy==p1r4) cout<<setw(12)<<p1;
            else cout<<setw(12)<<r2;
            cout<<endl<<endl;
        }
        for (ii=0;ii<nn;ii++) {
            dd=2*(r2-r1);
            ssf[ii]=0.97*ssf0[ii]*powl(dd/d0,-0.1);
            strssar=(stressr[ii][1]-stressr[ii][0])/2;
            strssat=(stresst[ii][1]-stresst[ii][0])/2;
            strssm=(stressr[ii][1]+stressr[ii][0])/2;
            strssm+=(stresst[ii][1]+stresst[ii][0])/2;
            strssa=sqrt(strssar*strssar+strssat*strssat-strssar*strssat);
            cc3=-200;
            ga[2+ii]=0;
            cc1=ssf[ii]*(ssf[ii]-ssc[ii])/(ssb[ii]+ssf[ii]);
            cc2=ssb[ii]*(sss[ii]-ssf[ii])/(ssb[ii]-ssf[ii]);

```

```

        if (-ssc[ii]<=strssm && strssm<cc1) cc3=ssc[ii]+strssm;
        if (cc1<=strssm && strssm<cc2) cc3=(1-strssm/ssb[ii])*ssf[ii];
        if (cc2<=strssm && strssm<sss[ii]) cc3=sss[ii]-strssm;
        if (cc3<1e16) ga[2+ii]=strssa-cc3;
    }
    ga[3]=50*(((r2*r2+r1*r1)/(r2*r2-r1*r1)+v1)*p1*r1/e1-ds);
}
// two-layer
if (nlayer==2) {
    if (yy==p1r4) ff=x[1];
    else ff=-x[1];
    if (yy==p1r4) r3=x[1];
    else p1=x[1];
    ga[0]=-x[0]+r1+rmin[0];
    ga[1]=-r3+x[0]+rmin[1];
    ga[2]=x[0]-r1-rmax[0];
    ga[3]=r3-x[0]-rmax[1];
    ga[4]=-x[2];
    p2=x[2];
    aa1=1/(x[0]*x[0])-1/(r1*r1);
    bb1=1/(r3*r3)-1/(x[0]*x[0]);
    cc1=p1;
    aa2=-((1+v1)/x[0]+(1-v1)*x[0]/(r1*r1))/e1;
    bb2=((1+v2)/x[0]+(1-v2)*x[0]/(r3*r3))/e2;
    cc2=(1-v1)*x[0]*p1/e1;
    cc=aa1*bb2-aa2*bb1;
    a1=(cc1*bb2-cc2*bb1)/cc;
    a2=(aa1*cc2-aa2*cc1)/cc;
    c1=-a1/(r1*r1)-p1;
    c2=-a2/(r3*r3);
    // inner surface stress of each layer under the press-fitted
    // and loading situation
    stressr[0][0]=0;
    stressr[0][1]=-p1;
    stresst[0][0]=-2*x[0]*x[0]*p2/(x[0]*x[0]-r1*r1);
    stresst[0][1]=-a1/(r1*r1)+c1+stresst[0][0];
    stressr[1][0]=-p2;
    stressr[1][1]=(a2/(x[0]*x[0])+c2)+stressr[1][0];
    stresst[1][0]=(r3*r3+x[0]*x[0])/(r3*r3-x[0]*x[0])*p2;
    stresst[1][1]=-a2/(x[0]*x[0])+c2+stresst[1][0];
}
if (result) {
    cout<<setw(12)<<"r2";
    if (yy==p1r4) cout<<setw(12)<<"r3";
    else cout<<setw(12)<<"p1";
    cout<<setw(12)<<"p2";
}

```

```

if (yy==p1r4) cout<<setw(12)<<"p1";
else cout<<setw(12)<<"r3";
cout<<setw(12)<<"pp2"<<setw(12)<<"dr2"<<endl;
for (ii=0;ii<nn;ii++) {
    cout<<setw(12)<<x[ii];
}
if (yy==p1r4) cout<<setw(12)<<p1;
else cout<<setw(12)<<r3;
cout<<setw(12)<<-stressr[1][1];
dd3=((x[0]*x[0]+r1*r1)/(x[0]*x[0]-r1*r1)-v1)/e1;
dd3+=((r3*r3+x[0]*x[0])/(r3*r3-x[0]*x[0])+v2)/e2;
cout<<setw(12)<<dd3*x[0]*p2<<endl<<endl;
}
// verify fatigue strenth of each layer
for (ii=0;ii<2;ii++) {
    if (ii==0) dd=2*(x[0]-r1);
    if (ii==1) dd=2*(r3-x[0]);
    ssf[ii]=0.97*ssf0[ii]*powl(dd/d0,-0.1);
    strssar=(stressr[ii][1]-stressr[ii][0])/2;
    strssat=(stresst[ii][1]-stresst[ii][0])/2;
    strssm=(stressr[ii][1]+stressr[ii][0])/2;
    strssm+=(stresst[ii][1]+stresst[ii][0])/2;
    strssa=sqrt(strssar*strssar+strssat*strssat-strssar*strssat);
    cc3=-200;
    ga[5+ii]=0;
    cc1=ssb[ii]*(ssf[ii]-ssc[ii])/(ssb[ii]+ssf[ii]);
    cc2=ssb[ii]*(sss[ii]-ssf[ii])/(ssb[ii]-ssf[ii]);
    if (-ssc[ii]<=strssm && strssm<cc1) cc3=ssc[ii]+strssm;
    if (cc1<=strssm && strssm<cc2) cc3=(1-strssm/ssb[ii])*ssf[ii];
    if (cc2<=strssm && strssm<sss[ii]) cc3=sss[ii]-strssm;
    if (cc3<1e16) ga[5+ii]=strssa-cc3;
}
ga[7]=50*(((1-v1)*c1*r1-(1+v1)*a1/r1)/e1-ds);
}
// three-layer
if (nlayer==3) {
    if (yy==p1r4) ff=x[2];
    else ff=-x[2];
    if (yy==p1r4) r4=x[2];
    else p1=x[2];
    ga[0]=-x[0]+r1+rmin[0];
    ga[1]=-x[1]+x[0]+rmin[1];
    ga[2]=-r4+x[1]+rmin[2];
    ga[3]=x[0]-r1-rmax[0];
    ga[4]=x[1]-x[0]-rmax[1];
}

```

```

ga[5]=r4-x[1]-rmax[2];
ga[6]=-x[3];
ga[7]=-x[4];
aa1=2*(r1*r1-x[0]*x[0]);
bb1=2*(x[0]*x[0]-x[1]*x[1]);
cc1=powl(x[1],2);
aa2=2*e2*((1+v1)*r1*r1+(1-v1)*x[0]*x[0]);
bb2=-2*e1*((1+v2)*x[1]*x[1]+(1-v2)*x[0]*x[0]);
cc2=(1+v2)*e1*powl(x[1],2);
cc=aa1*bb2-aa2*bb1;
c1=(cc1*bb2-cc2*bb1)/cc;
c2=(aa1*cc2-aa2*cc1)/cc;
a1=-2*c1*r1*r1;
a2=-powl(x[1],2)*(1+2*c2);
p2=x[3];
p3=x[4];
aa1=2*(x[0]*x[0]-r1*r1);
bb1=2*(x[1]*x[1]-x[0]*x[0]);
cc1=2*(r4*r4-x[1]*x[1]);
dd1=r1*r1*p1;
aa2=2*r1*r1*((1+v1)*e2-(1+v2)*e1)+2*x[0]*x[0]*((1+v2)*e1+(1-v1)*e2);
bb2=-4*e1*x[0]*x[0];
cc2=0;
dd2=r1*r1*((1+v2)*e1-(1+v1)*e2)*p1;
aa3=2*(1+v2)*e3*(r1*r1-x[0]*x[0]);
bb3=2*e3*((1+v2)*x[0]*x[0]+(1-v2)*x[1]*x[1]);
cc3=-2*e2*((1+v3)*r4*r4+(1-v3)*x[1]*x[1]);
dd3=-(1+v2)*e3*r1*r1*p1;
cc=aa1*bb2*cc3+aa2*bb3*cc1+aa3*bb1*cc2-aa1*bb3*cc2-aa2*bb1*cc3-aa3*bb2*cc1;
cp1=dd1*bb2*cc3+dd2*bb3*cc1+dd3*bb1*cc2-dd1*bb3*cc2-dd2*bb1*cc3-dd3*bb2*cc1;
cp1/=cc;
cp2=aa1*dd2*cc3+aa2*dd3*cc1+aa3*dd1*cc2-aa1*dd3*cc2-aa2*dd1*cc3-aa3*dd2*cc1;
cp2/=cc;
cp3=aa1*bb2*dd3+aa2*bb3*dd1+aa3*bb1*dd2-aa1*bb3*dd2-aa2*bb1*dd3-aa3*bb2*dd1;
cp3/=cc;
ap1=-r1*r1*(p1+2*cp1);
ap2=2*x[0]*x[0]*(cp1-cp2)+ap1;
ap3=-2*r4*r4*cp3;
// inner surface stress of each layer under the press-fitted
// and loading situation
stressr[0][0]=0;
stressr[0][1]=-p1;
stress[0][0]=(-a1/(r1*r1)+2*c1)*p3-2*x[0]*x[0]*p2/(x[0]*x[0]-r1*r1);
stress[0][1]=-ap1/(r1*r1)+2*cp1+stress[0][0];
stressr[1][0]=(a2/(x[0]*x[0])+2*c2)*p3-p2;

```

```

stressr[1][1]=ap2/(x[0]*x[0])+2*cp2+stressr[1][0];
stresst[1][0]=(-a2/(x[0]*x[0])+2*c2)*p3;
stresst[1][0]+=(x[1]*x[1]+x[0]*x[0])*p2/(x[1]*x[1]-x[0]*x[0]);
stresst[1][1]=-ap2/(x[0]*x[0])+2*cp2+stresst[1][0];
stressr[2][0]=-p3;
stressr[2][1]=ap3/(x[1]*x[1])+2*cp3-p3;
stresst[2][0]=(r4*r4+x[1]*x[1])*p3/(r4*r4-x[1]*x[1]);
stresst[2][1]=-ap3/(x[1]*x[1])+2*cp3+stresst[2][0];
if (result) {
    cout<<setw(10)<<"r2"<<setw(10)<<"r3";
    if (yy==p1r4) cout<<setw(10)<<"r4";
    else cout<<setw(10)<<"p1";
    cout<<setw(10)<<"p2"<<setw(10)<<"p3";
    if (yy==p1r4) cout<<setw(10)<<"p1";
    else cout<<setw(10)<<"r4";

    cout<<setw(10)<<"pp2"<<setw(10)
        <<"pp3"<<endl;
    for (ii=0;ii<nn;ii++) {
        cout<<setw(9)<<x[ii];
    }
    if (yy==p1r4) cout<<setw(15)<<p1;
    else cout<<setw(15)<<r4;
    cc1=(ap1+p3*a1)/(x[0]*x[0])+2*(cp1+p3*c1)-p2;
    cc2=ap3/(x[1]*x[1])+2*cp3-p3;
    cout<<setw(9)<<-cc1<<setw(9)<<-cc2<<endl<<endl;
    cc1=((x[0]*x[0]+r1*r1)/(x[0]*x[0]-r1*r1)-v1)/e1;
    cc1=x[0]*p2*(cc1+((x[1]*x[1]+x[0]*x[0])/(x[1]*x[1]-x[0]*x[0])+v2)/e2);
    cc2=-2*x[0]*x[0]*x[1]*p2/(e2*(x[1]*x[1]-x[0]*x[0]));
    cc2+=x[1]*p3*((r4*r4+x[1]*x[1])/(r4*r4-x[1]*x[1])+v3)/e3;
    cc2=-p3*(-(1+v2)*a2/x[1]+(1-v2)*c2*x[1])/e2;
    cout<<setw(10)<<"dr2"<<setw(10)<<"dr3"<<endl;
    cout<<setw(10)<<cc1<<setw(10)<<cc2<<endl;
}
for (ii=0;ii<3;ii++) {
    if (ii==0) dd=2*(x[0]-r1);
    if (ii==1) dd=2*(x[1]-x[0]);
    if (ii==2) dd=2*(r4-x[1]);
    ssf[ii]=0.97*ssf0[ii]*powl(dd/d0,-0.1);
    strssar=(stressr[ii][1]-stressr[ii][0])/2;
    strssat=(stresst[ii][1]-stresst[ii][0])/2;
    strssm=(stressr[ii][1]+stressr[ii][0])/2;
    strssm+=(stresst[ii][1]+stresst[ii][0])/2;
    strssa=sqrt(strssar*strssar+strssat*strssat-strssar*strssat);
    cc3=-200;
}

```



```

        ga[8+ii]=0;
        cc1=ssb[ii]*(ssf[ii]-ssc[ii])/(ssb[ii]+ssf[ii]);
        cc2=ssb[ii]*(sss[ii]-ssf[ii])/(ssb[ii]-ssf[ii]);
        if (-ssc[ii]<=strssm && strssm<cc1) cc3=ssc[ii]+strssm;
        if (cc1<=strssm && strssm<cc2) cc3=(1-strssm/ssb[ii])*ssf[ii];
        if (cc2<=strssm && strssm<sss[ii]) cc3=sss[ii]-strssm;
        if (cc3<1e16) ga[8+ii]=strssa-cc3;
    }
    ga[11]=50*((2*(1-v1)*cp1*r1-(1+v1)*ap1/r1)/e1-ds);
}
for (ii=0;ii<mm;ii++)
    if (ga[ii]>0) kk=1;
    return kk;
}
double function() {
    double ff;
    if (yy==p1r4) ff=x[nlayer-1];
    else ff=-x[nlayer-1];
    return ff;
}
main() {
    int i,j,k,k1,nh,nl,nc,count=0;
    double a,df;
    char y;
    const char convex='y';
    double xx[n2][nn],xo[nn],f[n2];
    for (i=0;i<3;i++) {
        sss[i]/=csafe;
        ssc[i]/=csafe;
        ssb[i]/=csafe;
        ssf0[i]=ssf0[i]*suf[i]/csafe;
    }
    cout<<" The number of the layer      ";
    cin>>nlayer;
    cout<<"Do you want to min r or max p1 in the optimum design?"<<endl
        <<"If you select to min r, please put r      ";
    cin>>yy;
    yy=tolower(yy);
    cout<<setiosflags(ios::left);
    cout<<"If feasible set is not convex, "
        <<"complex method is used. Please input y"<<endl;
    cin>>y;
    y=tolower(y);
    if (subject()) {
        for (i=0;i<mm;i++) {

```

```

        if (i<9) cout<<"g["<<(i+1)<<"]="<<setw(15)<<ga[i];
        else cout<<"g["<<(i+1)<<"]="<<setw(14)<<ga[i];
        if ((i+1)%4==0) cout<<endl;
    }
    cout<<"X0 is not feasible "; return 0;}
cout<<setiosflags(ios::left);
result=0;
loop1:
    for (i=0;i<nn;i++)
        xx[0][i]=x[i];
    for (i=1;i<n2;i++) {
        for (j=0;j<nn;j++) {
            xx[i][j]=bo[j][0]+rand()/32767.*(bo[j][1]-bo[j][0]);
            x[j]=xx[i][j];
        }
        if (!subject()) continue;
        for (j=0;j<nn;j++) {
            xo[j]=0.0;
            for (k=0;k<i;k++)
                xo[j]+=xx[k][j];
            xo[j]=xo[j]/i;
            x[j]=xo[j];
        }
        if (subject()) {
            if (y!=convex) {cout<<"Feasible is not convex. "<<endl; return 0;}
        }
        do{
            for (j=0;j<nn;j++) {
                xx[i][j]=xo[j]+0.5*(xx[i][j]-xo[j]);
                x[j]=xx[i][j];
            }
        } while (subject());
    }
loop2:
    for (i=0;i<n2;i++) {
        for (j=0;j<nn;j++)
            x[j]=xx[i][j];
        f[i]=function();
    }
loop3:
    nh=0;
    nl=0;
    for (i=0;i<n2;i++) {
        if (f[i]>f[nh]) nh=i;
        if (f[i]<f[nl]) nl=i;
    }

```

```

    }
    cout<<"fmax="<<setw(15)<<f[nh]<<"fmin="<<setw(15)<<f[nl];
    df=0.0;
    for (i=0;i<n2;i++)
        df+=powl((f[i]-f[nl]),2);
    df=sqrt(df/(n2-1));
    if (fabs(f[nl])>10.0) df/=fabs(f[nl]);
    count+=1;
    if (df<ee || count==20000*nn) {
        cout<<endl<<endl<<endl<<"The last results"<<endl<<endl;
/*
        for (i=0;i<nn;i++) {
            if (i<9) cout<<"X["<<(i+1)<<"]="<<setw(15)<<xx[nl][i];
            else cout<<"X["<<(i+1)<<"]="<<setw(14)<<xx[nl][i];
        }
        cout<<"f(X*)="<<f[nl]<<endl<<endl; */
        result=1;
        for (i=0;i<nn;i++)
            x[i]=xx[nl][i];
        cout<<"optimization result and relative parameter"<<endl;
        subject();
        cout<<endl<<"subject to "<<endl;
        for (i=0;i<mm;i++) {
            if (i<9) cout<<"g["<<(i+1)<<"]="<<setw(15)<<ga[i];
            else cout<<"g["<<(i+1)<<"]="<<setw(14)<<ga[i];
        }
        cout<<endl<<endl;
        return 0;
    }
    for (i=0;i<nn;i++) {
        xo[i]=0.0;
        for (j=0;j<n2;j++)
            xo[i]+=xx[j][i];
        xo[i]=(xo[i]-xx[nh][i])/(n2-1);
        x[i]=xo[i];
    }
    if (subject()) {
        if (y!=convex) {cout<<"Feasible is not convex. "<<endl; return 0;}
        for (i=0;i<nn;i++) {
            x[i]=xx[nl][i];
            bo[i][0]=xx[nl][i];
            if (fabs(xx[nh][i]-xx[nl][i])>2) bo[i][1]=xx[nh][i];
            else bo[i][1]=bo[i][0]+2;
        }
        goto loop1;
    }
}

```

```

loop4:
    a=1.5;
loop5:
    do {
        for (i=0;i<nn;i++)
            x[i]=xo[i]+a*(xo[i]-xx[nh][i]);
        a/=2.;
    } while (subject());
    if (function()<f[nh]) goto loop7;
    if (a>eb) goto loop5;
    k1+=1;
    if (k1==2) goto loop6;
    if (nh==0) nc=1; else nc=0;
    for (i=0;i<n2;i++)
        if ((i!=nh) && (f[i]>f[nc])) nc=i;
    for (i=0;i<nn;i++)
        xx[nh][i]=xx[nc][i];
    f[nh]=f[nc];
    goto loop4;
loop6:
    for (i=0;i<n2;i++) {
        for (j=0;j<nn;j++) {
            xx[i][j]=xx[nl][j]+0.5*(xx[i][j]-xx[nl][j]);
            x[j]=xx[i][j];
        }
        if (!subject()) continue;
        if (y!=convex) {cout<<"Feasible is not convex. "<<endl; return 0;}
        for (j=0;j<nn;j++) {
            x[j]=xx[nl][j];
            bo[j][0]=xx[nl][j];
            if (fabs(xx[nh][j]-xx[nl][j])>2) bo[j][1]=xx[nh][j];
            else bo[j][1]=bo[j][0]+2;
        }
        goto loop1;
    }
    k1=0; goto loop2;
loop7:
    k1=0;
    f[nh]=function();
    for (i=0;i<nn;i++)
        xx[nh][i]=x[i];
    goto loop3;
}

```

2. 程序使用输入数据说明

double sss[3]={2000,2000,2000};

sss[3] — 组合凹模 1~3 层材料屈服强度 σ_s , 按第 1 层 (内层)、中层和外层的顺序排列 (下同)

double ssc[3]={2400,2400,2400};

ssc[3] — 组合凹模 1~3 层材料压缩屈服强度 σ'_s

double ssb[3]={2600,2600,2600};

ssb[3] — 组合凹模 1~3 层材料拉伸极限强度 σ_b

double ssf0[3]={1200,1200,1200};

ssf0[3] — 组合凹模 1~3 层材料标准试件的疲劳强度 σ_{-1}

double suf[3]={0.92,0.92,0.92};

suf[3] — 组合凹模 1~3 层表面加工质量系数 β

double rmin[3]={10,10,10};

rmin[3] — 组合凹模 1~3 层最小壁厚 Δr_{\min}

double rmax[3]={880,880,880};

rmax[3] — 组合凹模 1~3 层最大壁厚 Δr_{\max}

double e1=210000,e2=210000,e3=210000;

ei — 组合凹模 1~3 层材料的弹性模量 $Ei(i=1,2,3)$

double v1=0.3,v2=0.3,v3=0.3;

vi — 组合凹模 1~3 层材料的泊桑系数 $vi(i=1,2,3)$

double ds=0.8;

ds — 组合凹模的刚度条件, 即内层允许的径向位移 $[\delta]$

double r1=70,r4=650,r3=280,r2=500,p1=1200,csafe=1.2,d0=9;

r1 — 组合凹模的内半径 r_1

r4 — 3 层组合凹模的外半径 r_4

r3 — 2 层组合凹模的外半径 r_3

r2 — 整体式凹模的外半径 r_2

p1 — 组合凹模预定的工作压力 p_1

csafe — 安全系数 n

d0 — 材料疲劳强度试件的直径 d_0

// one-layer data

// const int nn=1,mm=4;

nn 和 mm — 分别为整体凹模的变量数和约束条件数

// double x[nn]={350};

x[nn] — 整体凹模优化设计变量的初始点

// double bo[nn][2]={ {50,700} };

bo[nn][2] — 整体凹模优化设计变量的下限和上限

// two-layer data

// const int nn=3,mm=8;

nn 和 mm — 分别为 2 层组合凹模的变量数和约束条件数

// double x[nn]={97,560,229};

x[nn] — 2 层组合凹模优化设计变量的初始点

// double bo[nn][2]={ {68,140}, {80,200}, {200,400} };

bo[nn][2] — 2 层组合凹模优化设计变量的下限和上限

```
// three-layer data
const nn=5,mm=12;
double x[nn]={80,120,700,32,195};
double bo[nn][2]={ {68,140}, {90,250}, {90,800}, {50,600}, {50,400} };

```

nn 和 mm — 分别为 3 层组合凹模的变量数和约束条件数
x[nn] — 3 层组合凹模优化设计变量的初始点
bo[nn][2] — 3 层组合凹模优化设计变量的下限和上限

注：并不是所有数据都要输入，根据具体情况进行输入。如果 3 层组合凹模按给定组合凹模的承载能力，使组合凹模外半径最小的第一类目标函数进行优化设计，r4、r3 和 r2 等不必输入。

3. 程序运行中需要输入的数据说明

```
cout<<" The number of the layer ";
cin>>nlayer;
cout<<"Do you want to min r or max p1 in the optimum design?"<<endl;
<<"If you select to min r, please put r ";
cin>>yy;
cout<<"If feasible set is not convex, "
<<"complex method is used. Please input y"<<endl;
cin>>y;
```

nlayer — 组合凹模的层数
yy — 如果组合凹模的优化设计是使组合凹模外半径最小，则输入英文字母 r 或 R；如果组合凹模的优化设计是使组合凹模承载能力最大，则输入其他任意英文字母
y — 如果组合凹模疲劳强度优化设计的可行集为非凸集，继续进行优化计算，则输入英文字母 y 或 Y；如果组合凹模疲劳强度优化设计的可行集为非凸集，停止进行优化计算，则输入其他任意英文字母

4. 程序运行中打印结果说明

```
cout<<"fmax="<<setw(15)<<f[nh]<<"fmin="<<setw(15)<<f[nl];
cout<<setw(10)<<"r2"<<setw(10)<<"r3";
if (yy==p1r4) cout<<setw(10)<<"r4";
cout<<setw(10)<<"dr2"<<setw(10)<<"dr3"<<endl;
cout<<setw(10)<<"p1";
cout<<setw(10)<<"p2"<<setw(10)<<"p3";
cout<<setw(10)<<"pp2"<<setw(10)<<"pp3"<<endl;
```

fmax 和 fmin — 打印中间结果，复合形中，最坏点和最好点的函数值
r2、r3 和 r4 — 优化的组合凹模 1~3 层外半径，即 r_2 、 r_3 和 r_4
dr2 和 dr3 — 第 1 层与第 2 层和第 2 层与第 3 层之间的过盈量 Δr_2 和 Δr_3
p1 — 组合凹模承受的工作压力
p2 和 p3 — 组合凹模第 1 层与第 2 层和第 2 层与第 3 层之间的预紧装配后的径向应力
pp2 和 pp3 — 组合凹模第 1 层与第 2 层和第 2 层与第 3 层之间的工作时径向应力等

参 考 文 献

- [1] S.铁摩辛柯著. 张福范译.弹性稳定理论[M]. 北京: 科学出版社, 1958
- [2] 蒋尔雄等编著. 线性代数[M]. 北京: 人民教育出版社, 1981
- [3] Morhtar S. Bazaraa and C.M.Shetty.Nonlinear Programming Theory and Algorithms[M]. John Wiley Sons. 1979
- [4] 薛嘉庆编著. 最优化原理与方法[M]. 北京: 冶金工业出版社, 1983
- [5] 南京大学数学系编著. 最优化方法[M]. 北京: 科学出版社, 1978
- [6] 陈立周, 张英会, 吴清一等编著. 机械优化设计[M]. 上海: 上海科学技术出版社, 1983
- [7] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculation derivatives[J]. Computer J. 7, 155-162, 1964
- [8] 吴方. 关于 Powell 方法的一个注[J]. 数学学报, 20 (1) 1977
- [9] D. M. 希梅尔布劳著. 张义桑译. 实用非线性规划[M]. 北京: 科学出版社, 1981
- [10] David G. Luenberger. Linear and nonlinear programming[M]. Reading, Mass. Addison-Wesley, 1984
- [11] Garret N. Vanderplaats. Numerical optimization techniques for engineering design[M]. New York: McGraw-Hill, 1984
- [12] 詹姆斯恩.西多著. 应锦春译. 最优工程设计. 原理与应用[M]. 北京: 机械工业出版社, 1987
- [13] 王德人编著. 非线性方程组解法与最优化方法[M]. 北京: 人民教育出版社, 1979
- [14] 罗中华, 李社环一种求解最优化问题可行解的快速有效方法[J]. 湘潭大学学报, 1995 17 (3) :8-9
- [15] 罗中华, 李社环. 半惩罚函数法与内滑块十杆机构的优化设计[J]. 重型机械, 1998 (6) :31-33
- [16] Rochafeelar R. T.The multiplier method of Hestenes and Powell applied to convex programming[J]. J. Optimum theory application 1973, 12 (6) :555-562
- [17] 思科产品研发中心编著.MATLAB6.5 辅助优化计算与设计[M]. 北京: 电子工业出版社, 2003
- [18] 王小平, 曹立明著. 遗传算法 — 理论、应用与软件实现[M]. 西安: 西安交通大学出版社, 2002
- [19] 陈伦军, 罗延科, 陈海虹等编著. 机械优化设计遗传算法[M]. 北京: 机械工业出版社, 2005
- [20] 阎平凡, 张长水编著. 人工神经网络与模拟进化计算[M]. 北京: 清华大学出版社,

2000

[21] Zhonghua Luo and Zhiliang Zhang. Optimum design of fatigue strength of a cold extrusion compound container[J]. Journal of Manufacturing Science and Engineering 127 (1), 2005, 227-230

[22] 罗中华, 张质良. 多层压配组合冷挤压凹模疲劳强度优化设计[J]. 上海交大学报, 2002, 36 (4): 446-449

[23] 罗中华, 张质良. 多层压配组合冷挤压凹模的优化设计[J]. 塑性工程学报, 2003, 10 (4): 38-41

[24] Adel S Saada. Elasticity Theory and Application[M]. Pergamon Press Inc., N. Y., 1974, 323~337

[25] 河本實ほか共著. 金属の疲れと設計[M]. 日本: コロナ社, 昭和 47 年. 37-43

[26] 赵少汴, 王忠保主编. 疲劳设计[M]. 北京: 机械工业出版社, 1992. 39-40

[27] Fuchs H O, and Stephens R I. Metal Fatigue in Engineering[M]. New York: A Wiley-Interscience Publication, 1980. 176-181

[28] 罗中华. 冲裁零件排样优化[J]. 锻压技术, 1994 (3): 20-22

[29] 王孝培主编. 冲压手册[M] (第二版). 北京: 机械工业出版社, 2000

[30] 罗中华, 彭炎荣. 飞剪机剪切机构的优化设计[J]. 湘潭大学学报, 1996 18 (4): 73-77

[31] 邹家祥主编. 轧钢机械[M] (第三版). 北京: 冶金工业出版社, 2005

[32] 陈立周, 张英会, 吴清一等. 机械优化设计[M]. 上海: 上海科学技术出版社, 1987

[33] 罗中华. 蟹爪式装载机扒取机构的优化设计[J]. 湘潭大学学报, 1989 11 (3): 95-100

[34] 罗中华. 热锯机送进机构的优化设计[J], 重型机型, 1998, (5): 50-52

[35] 高满旭. 热锯机送进机构的优化设计[J], 重型机型, 1995, (6): 25-29

[36] 西北工业大学机械原理及机械零件教研室编. 机械设计 上册[M]. 北京: 人民教育出版社, 1979

[37] 洪慎章编著. 冷挤压实用技术[M]. 北京: 机械工业出版社, 2006

[38] 罗中华. J88 型冷挤压机拉力肘杆机构构件尺寸参数的确定[J]. 锻压机械, 1997, (2): 23-25

[39] 罗中华, 彭炎荣, 李社环等. J88 型冷挤压机拉力肘杆机构的优化设计[J]. 机械设计与研究, 1998, (2): 58-60

[40] 谢关烜. J88 型冷挤压机拉力肘杆机构的设计[J]. 锻压机械, 1978, (4): 2-12

[41] 何德誉主编. 曲柄压力机[M]. 北京: 机械工业出版社, 1981

[42] 罗中华, 刘国生. JA88 型冷挤压机压拉力肘杆机构的优化设计[J]. 湘潭大学学报, 1996, 18 (1): 106-110

[43] 罗中华, 彭炎荣. 双动拉延压力机外滑块传动机构的优化设计[J]. 锻压技术, 1997, 22 (3): 46-49

[44] 罗中华, 彭炎荣. 八连杆板冲压力机内外滑块执行机构的优化设计[J]. 锻压技术, 1997, 23 (5): 47-50

[45] 罗中华, 梅亚夫, 郭燕伶. 八连杆压力机执行机构部分尺寸参数的确定[J]. 湘潭大学学报, 1998, 20 (2): 79-82

- [46] В.И.符拉索夫主编. 夏萼辉等译. 锻造冲压曲柄压力机[M]. 上海: 上海科学文献出版社, 1982
- [47] 罗中华. 双动拉延压力机内滑块十杆机构的优化设计[J]. 机械设计, 1999, (1): 41-43
- [48] 华中工学院等五院(校)《机械传动及曲柄压力机》编写组编. 机械传动及曲柄压力机下册[M]. 北京: 人民教育出版社, 1978 年
- [49] 罗中华, 罗文波, 彭炎荣. 一种高精度有心扇形滑移线场的近似解析解. 锻压技术, 1999, (1): 43-45
- [50] 雷玉成, 汪建敏, 贾志宏主编. 金属材料成型原理[M]. 北京: 化学工业出版社, 2006
- [51] А.Д. Томленов. Растягивающих Напряжениях в Металле Приковке и Поперечной Прокатке. Изд Академии Наук СССР, 1960
- [52] 王祖唐等. 金属塑性成形原理[M]. 北京: 机械工业出版社, 1989
- [53] 罗中华, 梅亚夫, 彭炎荣等. 假塑性塑料熔体在鱼尾形流道中流动公式的推导[J]. 塑性工程学报, 1998, 5 (1), 61-65
- [54] 曹宏深, 赵仲治. 塑料成型工艺与模具设计[M]. 北京: 机械工业出版社, 1993
- [55] 唐志玉编著. 塑料模流变学设计[M]. 北京: 国防工业出版社, 1991